

UNIVERSIDADE PRESBITERIANA MACKENZIE

Jorge Cândido

**Detecção e Rastreamento de Faces
Utilizando Redes Bayesianas**

São Paulo
2006

Livros Grátis

<http://www.livrosgratis.com.br>

Milhares de livros grátis para download.

Jorge Cândido Detecção e Rastreamento de Faces Utilizando Redes Bayesianas 2006

JORGE CÂNDIDO

DETECÇÃO E RASTREIO DE FACES UTILIZANDO REDES
BAYESIANAS

**Dissertação apresentada à
Universidade Presbiteriana Mackenzie
como requisito parcial para obtenção
do título de Mestre em Engenharia
Elétrica.**

ORIENTADOR: Prof. Dr. MAURÍCIO MARENGONI

São Paulo
2006

Cândido, Jorge

Detecção e Rastreamento de Faces Utilizando Redes Bayesianas / Jorge Cândido. - 2006.

39 f. ; 30 cm.

Orientador: Mauricio Marengoni.

Dissertação (Mestrado em Engenharia Elétrica)-
Universidade Presbiteriana Mackenzie, São Paulo, 2006.

Bibliografia: f. 83-87.

1. Visão Computacional
2. Redes Bayesianas
3. Processamento de Imagens. I. Título.

JORGE CÂNDIDO

**DETECÇÃO E RASTREIO DE FACES
UTILIZANDO REDES BAYESIANAS**

Dissertação apresentada à Universidade
Presbiteriana Mackenzie como requisito
parcial para obtenção do título de
Mestre em Engenharia Elétrica.

Aprovado em Fevereiro de 2006

BANCA EXAMINADORA

Prof. Dr. Mauricio Marengoni

Prof. Dr. Carlos Henrique Costa Ribeiro

Prof. Dr. Luiz Henrique Alves Monteiro

À minha esposa, pelo apoio incondicional, fundamental nos momentos em que os objetivos pareciam mais distantes.

AGRADECIMENTOS

Ao Professor Mauricio Marengoni, o qual esteve sempre presente desde os passos iniciais deste projeto, incentivando e cobrando quando necessário.

À Professora Pollyana Notargiocomo Mustaro, que esclareceu muitas dúvidas durante o processo de elaboração gráfica deste trabalho.

A todos os professores e funcionários da Pós-Graduação Mackenzie, pelo incentivo à realização deste trabalho.

A Peter Peer, pela permissão de uso das imagens do CVL FACE DATABASE.

A todas as pessoas que permitiram que suas fotos fossem usadas durante este trabalho.

Aos espíritos iluminados do plano superior, que estiveram sempre ao meu lado e prontos para me orientar e socorrer.

RESUMO

Neste trabalho, foi desenvolvido um sistema automático de detecção e rastreamento de face. O sistema de detecção de face é composto de três operadores visuais que fazem a detecção de atributos faciais simples. A característica principal destes operadores é que eles são construídos para funcionar com custo computacional baixo, otimizando o tempo de processamento. Os operadores foram construídos utilizando técnicas de redes neurais para a detecção dos olhos, casamento de modelos para a detecção da boca e para a identificação de cor de pele. Os operadores visuais formam os elementos básicos do detector de face cujos resultados são combinados em uma Rede Bayesiana que define o resultado final do algoritmo de detecção de face. O uso da Rede Bayesiana proporciona ao sistema um melhor desempenho em relação aos operadores individualmente. Este detector foi implementado de forma a analisar as imagens captadas quadro a quadro por uma câmera de vídeo USB. Quando acontece uma detecção de face, a mesma passa a ser rastreada por um sistema de rastreamento que busca a região de cor de pele da face, determinando sua posição na tela. Este sistema pode ser utilizado em diversas aplicações que requerem a detecção de faces, em particular, este sistema faz parte de um sistema de mouse óptico de alta precisão para, por exemplo, facilitar a interface homem máquina para pessoas com problemas de coordenação motora.

ABSTRACT

This work presents a face detection system that uses a Bayesian Network to combine information from different computational cheap visual operators. The aim in this work is to show that combining simple features in a Bayesian Network allows building an enhanced face detector system, increasing the detection rate and speeding up the face detection process. This face detector has been developed to work in the stream acquired by a USB computer camera. After the detector finds a face in the stream, a face tracking system begins to work, locating the eyes position at the screen. The system is part of a high precision optical mouse system that will be used in human-computer interface, helping users with certain disabilities.

SUMÁRIO

1	INTRODUÇÃO	11
1.1	Objetivos	18
2	FUNDAMENTOS TEÓRICOS	21
2.1	Detecção de Faces	21
2.1.1	Redes Neurais Artificiais.....	22
2.1.2	Métodos Baseados em Características Faciais.....	27
2.1.2.1	Paul Viola e Michael J.Jones.....	27
2.1.2.2	K. C. Yow e R. Cipolla.....	31
2.1.3	Modelos Flexíveis.....	33
2.2	Rastreo de Faces	37
2.2.1	Métodos usando Pontos de Interesse.....	37
2.2.2	Modelos Elásticos – Snakes.....	38
2.2.3	Algoritmo CAMSHIFT.....	40
2.3	Fusão de Informação	43
3	MODELAGEM	46
3.1	Operador Pele	48
3.2	Operador Olhos	49
3.3	Operador Boca	50
3.4	Rastreo da Face	53
4	EXPERIMENTOS E RESULTADOS	57
4.1	Banco de Imagens	57
4.2	Classificação de Cores	60
4.3	Operador Olhos	62
4.4	Rede Bayesiana	65
4.4.1	Tabelas de Probabilidades.....	66
4.4.2	Inferência da Rede.....	70
4.5	Detector de Faces	73
4.6	Rastreo de Face	79
5	CONCLUSÃO	81
6	REFERENCIA BIBLIOGRÁFICA	83

1. INTRODUÇÃO

Nas últimas décadas, a área de visão computacional tem atraído bastante a atenção dos pesquisadores dentro da comunidade de Inteligência Artificial. O constante aumento da capacidade de processamento dos computadores modernos permite que novas e promissoras tecnologias na área de visão computacional se tornem realidade. Ao mesmo tempo, a diminuição do preço dos dispositivos de captação de vídeo e imagens pode contribuir para a popularização de aplicações como sistemas de vigilância em circuito fechado, sistemas automáticos de inspeção de alimentos para consumo e busca em bancos de impressões digitais para agências governamentais [41].

Uma das tarefas principais em visão computacional é a detecção de objetos. A detecção de objetos é a primeira aproximação para muitas tarefas de visão, como por exemplo, reconhecimento de padrões, rastreamento e classificação de objetos. O que torna esta tarefa desafiadora é a grande variação que um determinado objeto pode apresentar em uma imagem. Variações que podem ser causadas por interferências externas (condições do ambiente e ruído), ou então variações que são intrínsecas do próprio objeto. Por exemplo, um algoritmo que se propõe a detectar automaticamente automóveis em imagens, tem que lidar com inúmeras possibilidades de formas e cores que os automóveis podem apresentar [35]. Algumas destas diferenças podem ser vistas nas imagens de carros da Figura 1.



Fig.1: Exemplos de imagens de carros.

O tema principal desta pesquisa é a detecção de faces. O aumento no número de pesquisas sobre a detecção de faces em imagens digitais é motivado pela premissa que os computadores podem extrair informações como identidade, situação e intenção, e agir conforme a situação [39]. Em consequência, grandes avanços têm sido feitos nesta área. No entanto, os algoritmos apresentados até hoje, invariavelmente apresentam ou falta de precisão e robustez ou demandam muito tempo de processamento para terem uma resposta precisa.

A definição clássica de detecção de faces é: dado uma imagem qualquer, determinar se existe ou não alguma face e

retornar a localização para cada face presente [39]. Esta tarefa é distinta da tarefa de reconhecimento de face. Reconhecimento de faces é a classificação de uma face segundo um banco de dados de faces conhecidas do sistema. Diferente de um sistema de detecção de faces que recebe como entrada uma imagem qualquer para ser analisada, um sistema de reconhecimento trabalha normalmente com uma pequena área da imagem contendo apenas uma face. Seu objetivo é apenas determinar a que indivíduo a face pertence. Portanto é natural que o reconhecimento de faces seja precedido de uma etapa de detecção. A Figura 2 ilustra a diferença entre estas duas tarefas.

Este projeto faz parte de um projeto já em andamento, que busca o desenvolvimento de um *mouse ótico* controlado pelo movimento dos olhos do operador. Esta tecnologia permitirá que usuários de computadores com dificuldades ou deficiências motoras (por exemplo, com doença de Parkinson), controlem o cursor na tela do computador, com precisão suficiente para permitir a navegação pelas telas e até mesmo, que clique em certas posições. Existem alguns sistemas similares já em funcionamento, mas em geral eles apresentam algumas limitações. Para exemplificar, a seguir estão descritos dois sistemas diferentes de mouse óticos.

O sistema mostrado na Figura 3 opera com uma câmera de vídeo que faz a leitura da posição para onde o olho da pessoa está olhando. A imagem do olho é projetada por uma lente especial transparente que fica montada entre o olho e o monitor, presa por uma estrutura à cabeça do operador. Uma fonte de luz infravermelha é aplicada diretamente no olho da pessoa e a pupila reflete a luz que é captada pela câmera, como indicado na Figura 4.

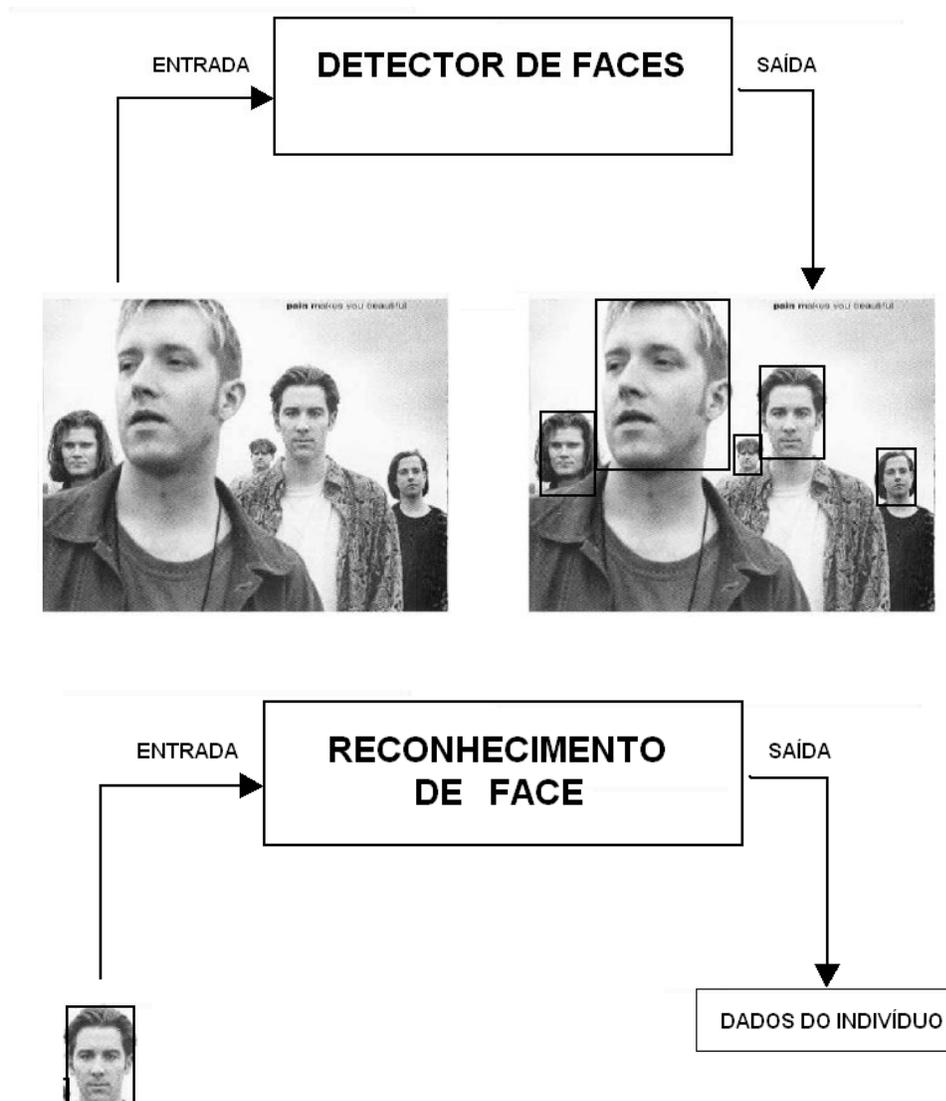


Fig.2: Comparação entre sistemas de detecção e reconhecimento de faces.

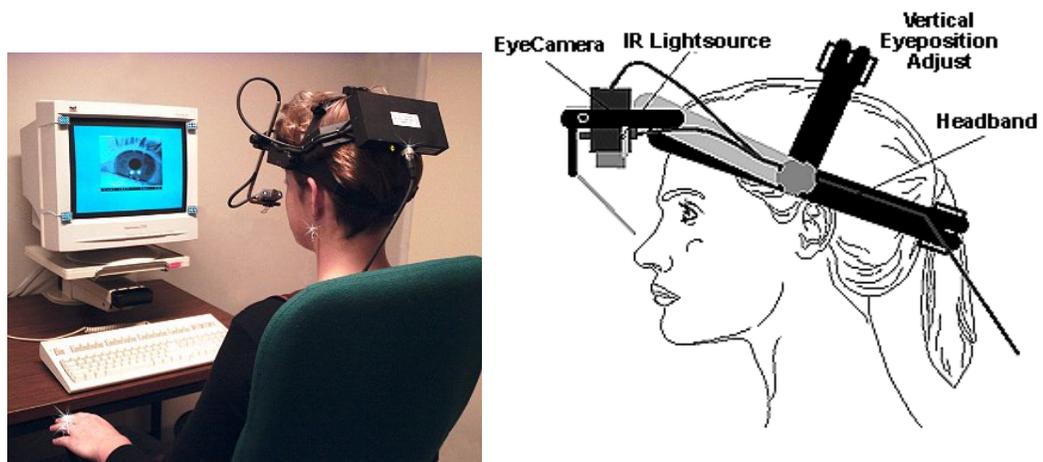


Fig.3: Mouse óptico com câmera presa à cabeça do operador.

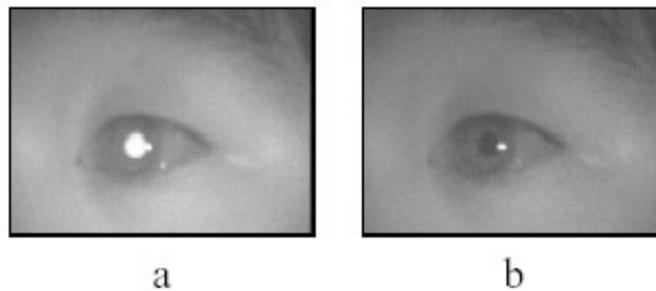


Fig.4: A luz refletida na pupila é facilmente rastreada pela câmera.

O sistema mostrado na Figura 5 funciona captando os impulsos elétricos dos músculos da face através de sensores de eletrocardiograma. Estes sensores ficam fixos em torno dos olhos e o sistema identifica o movimento do globo ocular do operador e calcula a posição na tela do computador para onde o operador está olhando.



Fig.5: Mouse óptico que opera captando o movimento dos músculos faciais.

Neste projeto, o objetivo é fazer uso de uma *WEBCAM* comum para capturar a imagem em frente da tela do computador e, através de um programa instalado no computador, determinar se existe uma pessoa olhando para a tela e estimar sua posição no campo visual da câmera.

A biblioteca de software básica, que faz a captura e manipulação das imagens da *WEBCAM*, é a *OpenCV* da Intel (*Open Source Computer Vision*). Esta biblioteca de funções programáveis foi desenvolvida nos laboratórios da Intel para uso, principalmente, em aplicações de visão computacional em tempo real. O manual de referencia do *OpenCV* [18] foi o documento principal de consulta da biblioteca usado neste projeto.

Assim que uma nova face é detectada na seqüência de imagens, sua localização é calculada. Por meio de um modelo geométrico, a possível área onde os olhos da pessoa estão localizados será passada para a próxima camada do sistema que fará a localização do ponto na tela do computador para onde a pessoa está olhando (visada) conforme esquematizado

na Figura 6. Uma vez que os olhos são detectados, um processo simples de calibração deverá proporcionar a precisão desejada. Este projeto se limitará ao desenvolvimento da ferramenta de detecção e rastreamento da face do operador sentado em frente ao computador, ou seja, a camada intermediária do projeto de mouse óptico mostrado na Figura 6.

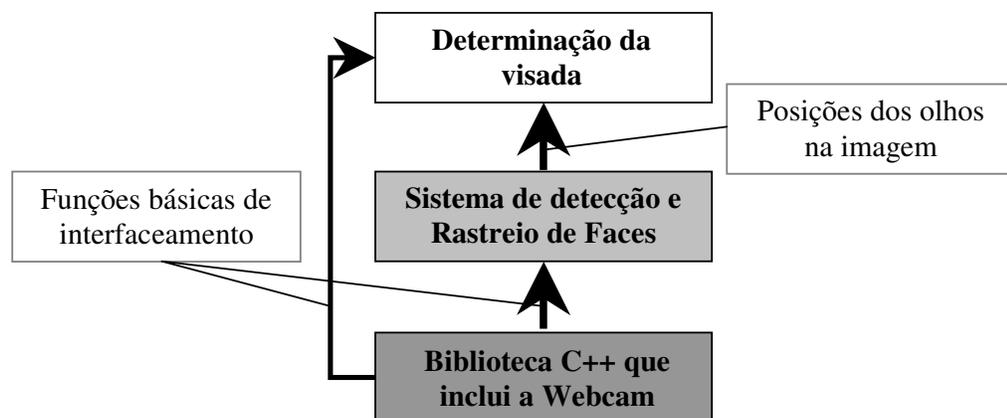


Fig.6: Camadas do projeto do mouse óptico e das relações entre as camadas do projeto.

O sistema de detecção aqui apresentado utiliza operadores simples (computacionalmente baratos) e criar uma estrutura de controle que permita rapidez e robustez no processo de detecção. Individualmente, cada detector de características faciais não fornece informação suficiente para decidir sobre a presença ou não de uma face na imagem. Porém, a fusão das informações fornecidas pelos operadores, melhora a qualidade da detecção e é rápido o bastante para permite o uso do sistema em tempo real. Esta abordagem foi utilizada no trabalho apresentado por Candido e Marengoni [1].

A idéia de se combinar múltiplos detectores de características faciais (*Multiple Features*) não é nova e já foi utilizado por Chen et al. [5] e Wu et al. [44]. A principal diferença do sistema aqui apresentado é a combinação dos resultados em uma estrutura de probabilidades do tipo Rede Bayesiana. Os experimentos realizados demonstram que a correta escolha dos operadores (detectores de características faciais) e estrutura da rede (arcos e nós), leva a um aumento da taxa de detecção diminuindo o número de falso-positivos.

1.1. Objetivos

O objetivo principal deste projeto é desenvolver um algoritmo de busca, detecção e rastreamento de faces de pessoas que estejam na frente de uma tela de computador. A captura da imagem que será analisada é feita por uma *WEBCAM*. Após a etapa de detecção, o algoritmo deverá permanecer rastreando e corrigindo a localização da face, até o momento que a pessoa saia do campo de visão da câmera, ou então o sistema não consiga mais detectar a face.

A seqüência de imagens apresentada na Figura 7 exemplifica uma operação completa do sistema de detecção de faces. No início, enquanto não há movimento na imagem (Figura 7a), o sistema permanece sem aplicar os operadores de características faciais. Nesta etapa a detecção de movimento é feita simplesmente por subtração de quadros em seqüência. Enquanto não ocorre movimento na imagem, a subtração de quadro resulta em uma matriz com todos os valores próximos de zero. Assim que um objeto entra no campo de visão da *WEBCAM*, o sistema começa a analisar a imagem para fazer a detecção (Figura 7b). Quando uma face é detectada (Figura 7c), a posição da face é passada para o

sistema de rastreamento de faces que determina a posição dos olhos de forma precisa e esta posição é passada para o sistema que irá fazer o controle de visada (Figura 7d). O sistema permanece fazendo o rastreamento da face e determinando a posição dos olhos até que o usuário saia da frente do campo visual da câmera ou se movimente além dos limites de rastreamento/detecção de faces (Figura 7e). Quando a imagem retorna a uma condição sem movimento, o sistema retorna a condição inicial.



Fig. 7: Seqüência de funcionamento do mouse ótico.

2. FUNDAMENTOS TEÓRICOS E TRABALHOS RELACIONADOS

Neste capítulo estão contidas as bases teóricas dos processos e métodos usados neste projeto.

Em primeiro lugar serão discutidas algumas técnicas usadas por outros autores na tarefa de detecção de faces. Em seguida são apresentados alguns trabalhos sobre o rastreamento de objetos em uma seqüência de imagens (*stream*). Por último temos uma pequena introdução teórica sobre Fusão de Informação, especificamente sobre a teoria de Redes Bayesianas e suas aplicações.

2.1. Detecção de Face

As pesquisas realizadas sobre o tema detecção de face resultaram em algumas soluções diferentes para este problema [39]. Algumas destas soluções realizam a busca por características locais para localizar a face. Estas características podem ser as mais diversas como, por exemplo, cor da pele, textura e características faciais como boca e olhos [44]. Outras soluções apresentam métodos de aprendizagem para treinar o algoritmo para detectar uma face em imagens. São exemplos deste tipo de abordagem as Redes Neurais [32] [33].

Na seqüência deste capítulo serão apresentadas, em detalhes, algumas abordagens que já foram utilizados na tarefa de detecção de face. Ao mesmo tempo, será dada a fundamentação teórica sobre as técnicas utilizadas nestes métodos.

2.1.1. Redes Neurais Artificiais

Os fundamentos de redes neurais artificiais surgiram em 1943. Neste ano McCulloch e Pitts [52] criaram um modelo matemático simples que representa as funções do que seria um neurônio artificial. Este modelo matemático é representado pelo desenho da Figura 8. Esta estrutura é a base de construção das redes neurais artificiais.

As redes neurais possuem habilidade para executar computação distribuída, tolerar entradas ruidosas e aprender segundo Russel e Norvig [34].

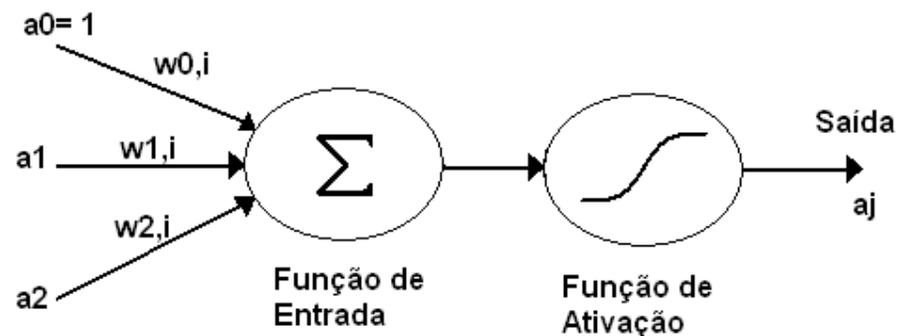


Fig. 8: Modelo matemático do neurônio.

Cada neurônio em uma rede neural é uma unidade ou nó. As unidades são vinculadas por um valor de peso $w(j, i)$. Desta forma a função de entrada realiza uma soma ponderada entre os valores de entrada $a(j)$. O sinal a_0 é chamado de peso de desvio (bias), e tem a função de ajustar o valor de limite da função de ativação.

Uma escolha possível para a função de ativação do neurônio é a função sigmóide (função logística) que tem a saída entre -1 e $+1$ [34]. O fato de a função sigmóide ser

diferenciável facilita bastante durante o processo de aprendizagem da rede neural.

Segundo sua estrutura, uma rede neural pode ser classificada como de alimentação direta (*feed-forward*) ou recorrente. Na rede neural de alimentação direta, o fluxo de informação segue apenas um sentido: da entrada para a saída. Em uma rede neural recorrente a informação tomada em uma camada pode ser aplicada em uma camada anterior. As Figuras 9 e 10 mostram os diagramas representando estas estruturas.

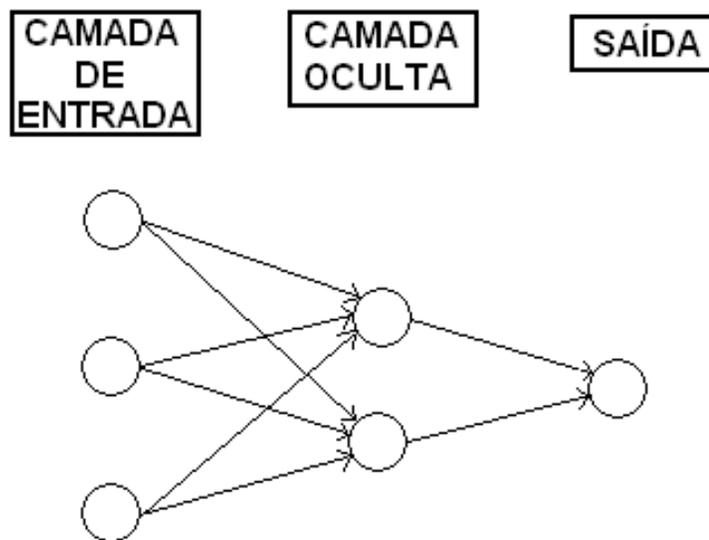


Fig. 9: Rede Neural de alimentação direta.

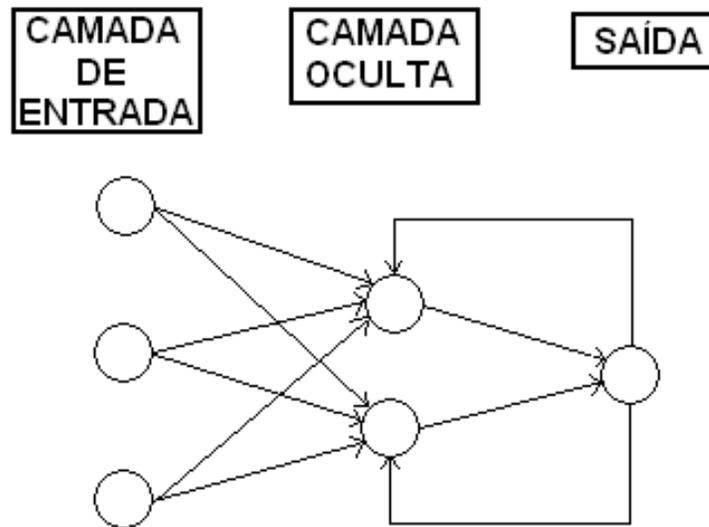


Fig. 10: Rede Neural recorrente.

Os neurônios de uma rede neural ficam divididos em camadas. As redes neurais possuem pelo menos a camada de entrada, representando os valores de entrada da rede, e uma camada de saída, onde a resposta é obtida. A rede neural pode ainda ter uma ou mais camadas intermediárias, também chamada de camada oculta. Quando a rede neural não possui camada alguma, ela é conhecida como *perceptron* [34].

O método de treinamento de rede supervisionado conhecido como propagação de retorno (*back-propagation*), envolve um processo iterativo de diminuição do erro na saída da rede. Através deste processo os exemplos de entrada são apresentados à camada de entrada e a saída correspondente é calculada. A cada nova saída que é calculada, o erro apresentado em relação a saída esperada, é usado para ajustar os valores dos pesos $w(i,j)$ com o objetivo de diminuir este erro. Uma explanação clara do processo de treinamento

de propagação de retorno pode ser encontrada em [34]. O processo de treinamento é interrompido quando se alcança o número máximo de iterações ou o erro atinge o valor desejado.

Detecção de objetos é uma aplicação típica para redes neurais. Rowley, Bajula e Kanade [32] apresentam um sistema de detecção de faces frontais com orientação vertical baseado em redes neurais.

Segundo os autores, o que dificulta a tarefa de identificação de faces em uma rede neural é o problema de caracterização da condição de “não-face”. O resultado do uso de um conjunto pequeno de amostras negativas durante o treinamento da rede, é um índice elevado de falso-positivos. Para minimizar este efeito, os autores citam um método iterativo de treinamento com os seguintes passos:

1. Etapa de treinamento da rede neural com 1000 imagens de faces escolhidas aleatoriamente do conjunto de treinamento;

2. Etapa de avaliação da rede neural com imagens de paisagens que não contém faces. São selecionadas, neste teste, as janelas de imagens que deram, erroneamente, resultado positivo;

3. A seguir é realizada uma nova etapa de treinamento da rede neural, utilizando as imagens selecionadas no item 2.

Para o treinamento da rede em imagens contendo faces, foram selecionadas 1050 imagens de bancos de dados disponíveis nas Universidades de Carnegie Mellon e Harvard.

Na etapa de treinamento, as imagens passaram por um processo de normalização de tamanho e padrão de

iluminação. Após a normalização, para cada imagem, foram geradas outras 15 amostras que sofreram aleatoriamente rotação de até 10° em qualquer direção, aumento ou redução do tamanho da imagem entre 90% e 110% e espelhamento. Estas amostras foram utilizadas no processo de aprendizado de forma a aumentar a robustez do sistema na identificação das faces.

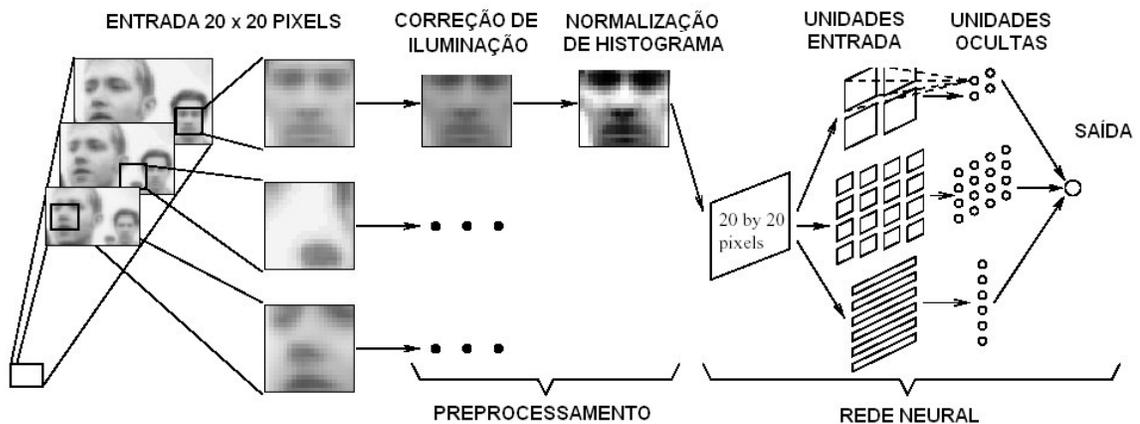


Fig.11: Esquema de rede neural para detecção de faces utilizada por Kanade et al. [32] e [33].

A camada de entrada da rede neural é composta por 400 neurônios que recebem os valores dos pixels em tons de cinza de uma janela de 20x20 da imagem. A camada de entrada alimenta três tipos de camadas ocultas: quatro que recebem as informações de quadros de 10x10, dezesseis que recebem as informações de quadros de 5x5 e seis que recebem as informações sobrepostas de faixas horizontais de 20x5 pixels conforme indicado na Figura 11.

Cada tipo de camada oculta opera como um especialista em detectar um tipo de característica facial. Por exemplo, as

faixas horizontais são melhores para detectar a simetria dos olhos.

O problema principal deste sistema é o tempo de processamento, já que todos os possíveis quadros de 20x20 existentes na imagem devem ser analisados [39]. Portanto é uma busca exaustiva e, dependendo do tamanho da imagem original, requer um alto tempo de processamento.

2.1.2. Métodos Baseados em Características Faciais

Métodos baseados em características faciais, em inglês "*feature-based methods*", são aqueles que fazem uso de algoritmos capazes de detectar algumas características faciais fixas como olhos, sobrancelha, boca e nariz, e baseado nestes detectores decidir sobre a presença de uma face ou não.

São também conhecidos por métodos "*botton-up*", porque analisam a imagem em sua estrutura mais básica (pixels) procurando por características locais para definir um objeto maior. Na seqüência serão descritos dois sistemas baseados neste método.

2.1.2.1. Paul Viola e Michael J. Jones

Viola e Jones descrevem em [38] um sistema que é capaz de detectar faces em posição frontal. Uma característica importante deste sistema é a velocidade de processamento das imagens. O sistema é capaz de realizar busca por faces em imagens com resolução de 384 por 288 pixels a uma taxa de 15 imagens por segundo rodando em um computador com processador Intel Pentium III de 700Mhz.

Este sistema introduz um novo modo de representar a imagem, chamado “*Integral Image*”. Este modo de representação será especialmente útil no processo de detecção das características geométricas (*features*).

Ao invés de representar a imagem pelo valor de intensidade dos pixels $i(x,y)$, o “*Integral Image*” usa a soma de todos os pixels acima e a esquerda $ii(x,y)$. A Figura 12 mostra uma área em que a soma dos pixels é realizada para o ponto (x,y) :

$$ii(x_p, y_q) = \sum_{q=0}^q \sum_{p=0}^p i(x_p, y_q) \quad (1)$$

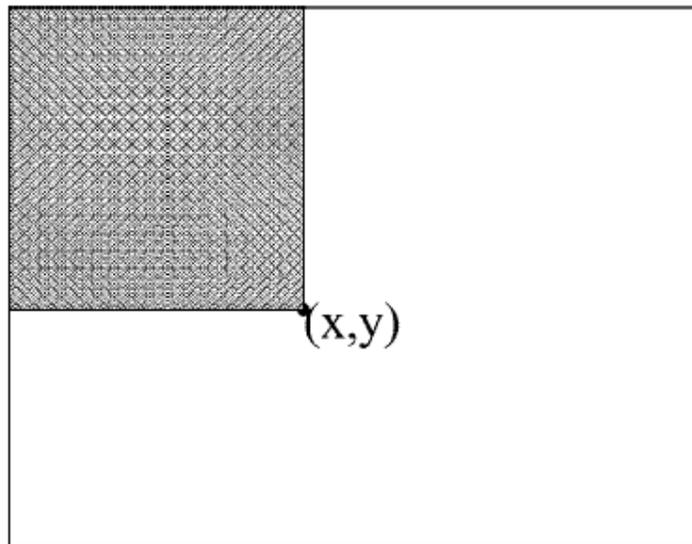


Fig.12: Representação da imagem pela soma da intensidade dos pixels acima e a esquerda de cada ponto.

Depois que a imagem inteira está codificada na representação descrita acima, o algoritmo faz a busca por certos atributos dentro da imagem. Na Figura 13 é mostrado o desenho básico para quatro atributos, sendo que o primeiro é formado de dois retângulos montados na vertical, o segundo com dois retângulos na horizontal, o terceiro com três retângulos e o quarto com quatro retângulos. O valor de cada atributos é igual à diferença entre a soma dos pixels dentro dos retângulos claro e escuro.

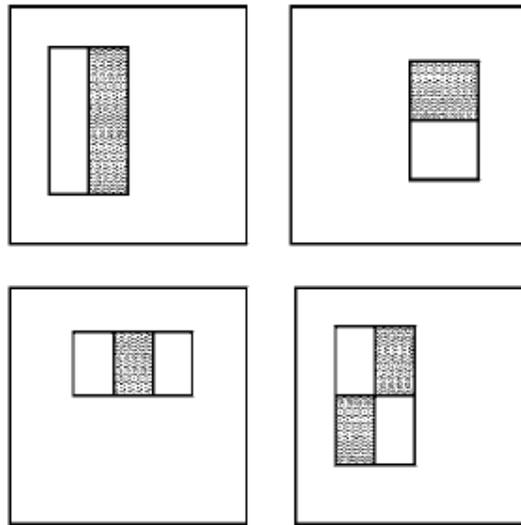


Fig.13: Representação geométrica das “features” usadas no sistema de Viola e Jones.

O uso da representação de imagem “*Integral Image*” facilita o cálculo da soma das intensidades de pixels dentro de qualquer retângulo. O resultado é obtido com apenas quatro valores tomados da representação “*Integral Image*”, como mostra a Figura 14. Isto é um dos fatores que torna o sistema rápido.

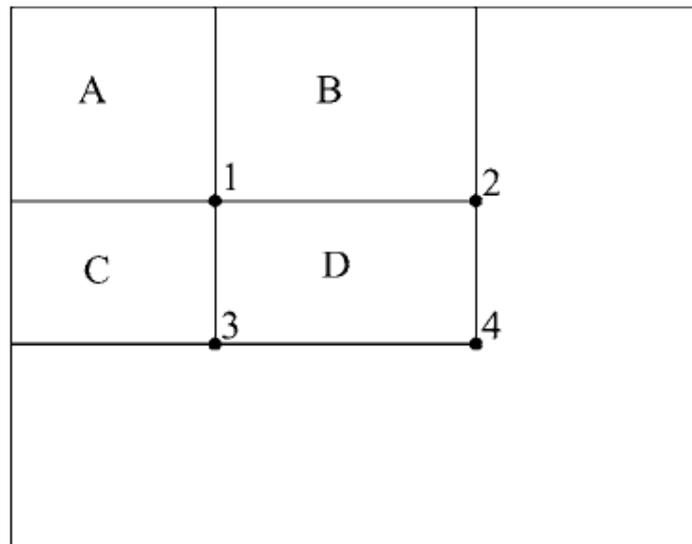


Fig. 14: Método de cálculo da soma dos pixels de um retângulo.

Outro fator que torna o sistema rápido é a escolha correta dos classificadores que são aplicados primeiro na imagem. Estes classificadores são aqueles que eliminam a maior parte das imagens que não são face. Desta forma, os classificadores mais complexos são aplicados somente em regiões da imagem com maior potencial de ser uma face.

Aplicando o algoritmo de aprendizagem ADABOOST apresentado por Freund e Schapire [14], foram escolhidos alguns dos atributos dentro da imagem para formar os classificadores que serão aplicados à imagem. O algoritmo ADABOOST é usado para selecionar os atributos e também para treinar os classificadores.

A estrutura final do detector é composta por 38 classificadores em cascata que usam no total 6060 atributos.

O primeiro classificador usa apenas dois atributos e é capaz de rejeitar 50% das imagens não-face e aceitar perto de 100% das imagens face. O próximo classificador usa dez atributos. Os próximos dois classificadores usam 25 atributos cada, e são seguidos por três classificadores com 50 atributos. Os próximos classificadores usam quantidades variadas de atributos.

2.1.2.2. Kin Choong Yow e Roberto Cipolla

Yow e Cipolla [41] mostram um método baseado em busca de características faciais locais usando padrões geométricos em níveis de cinza.

Depois de localizadas, as características faciais são combinadas em grupos (*“Partial Face Groups” - PFG*) que representam modelos para sobrelhas, olhos, nariz e boca. Estes grupos mantêm entre si certas relações geométricas e espaciais que são ditadas pelo conceito chamado de *“Perceptual Organization”* introduzido por Lowe [26], conforme apresentado na Figura 15. Este fenômeno, que tem sido estudado pelos campos da psicologia e visão computacional, permite que pessoas rapidamente identifiquem os diferentes objetos que vemos a partir de características como paralelismo, conectividade e padrões repetitivos.

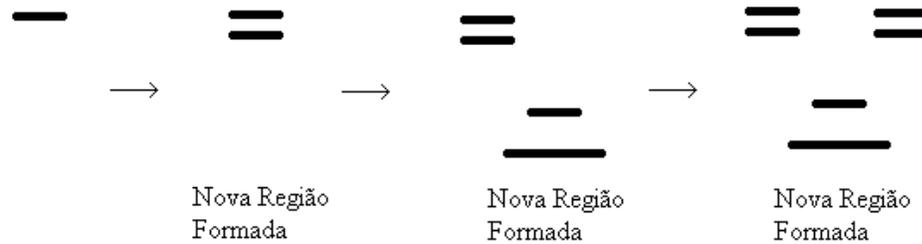


Fig. 15: Agrupamento progressivo das características faciais utilizados por Yow e Cipolla [41].

Antes de iniciar a busca, os pontos de interesse da imagem (que tem potencial para conter as características procuradas) são selecionados em uma etapa de pré-processamento. Quando as imagens são transformadas para baixa resolução, as características faciais procuradas têm aparência de barras alongadas escuras sobre um fundo claro. Por causa disto, a etapa de pré-processamento é formada por um filtro gaussiano de segunda ordem capaz de identificar formas alongadas. Na resposta do filtro, os pontos de máxima locais são selecionados como pontos de interesse onde as características faciais podem ser encontradas.

No final, os grupos de PFG's selecionados são combinados em uma estrutura probabilística com o objetivo de diminuir o número de falso-positivos. A estrutura probabilística é uma Rede Bayesiana que usa seis nós para caracterizar a face (Figura 16).

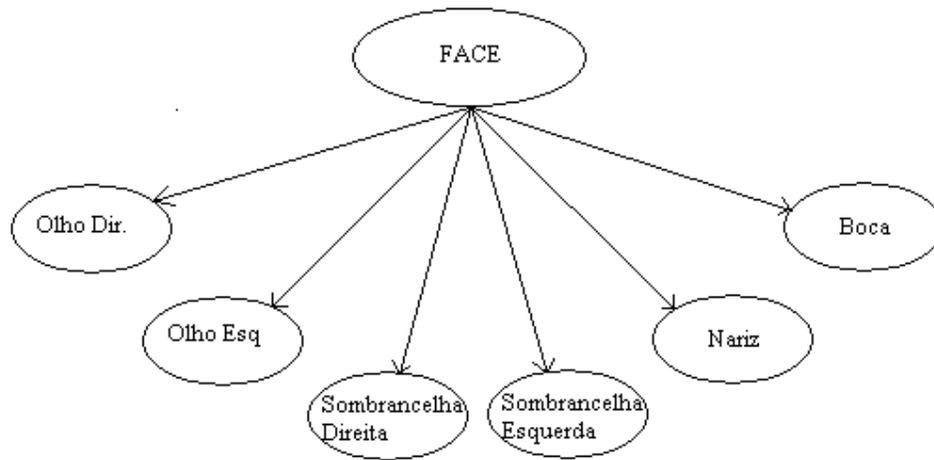


Fig. 16: Estrutura de Rede usada por Yow e Cipolla.

2.1.3. Modelos Flexíveis – ASM/AAM

Outra abordagem para a tarefa de detecção de faces é o uso de modelos flexíveis.

Um modelo flexível de pontos é um conjunto de coordenadas que indicam pontos relevantes (“*Landmarks*”) para o objeto dentro da imagem. O sistema de detecção baseado neste modelo funciona do seguinte modo: dada uma grosseira aproximação, este modelo pode se mover e combinar com uma nova imagem. São exemplos de uso de modelos elásticos os algoritmos ASM (“*Active Shape Model*”) [9] e AAP (“*Active Appearance Model*”) [11].

O modelo de pontos usado é capaz de ser transformado em rotação, translação e escala para aproximar-se da forma do objeto na imagem.

Durante o processo de busca pela melhor combinação entre os pontos do modelo e a imagem, ocorre o seguinte processo:

1. Exame da vizinhança de cada ponto para melhor combinação;
2. Atualização dos parâmetros do modelo para combinar os pontos;
3. Repetição até convergência.

Na prática os pontos para combinação são procurados ao longo de uma reta perpendicular ao contorno do modelo, como mostra a Figura 17.

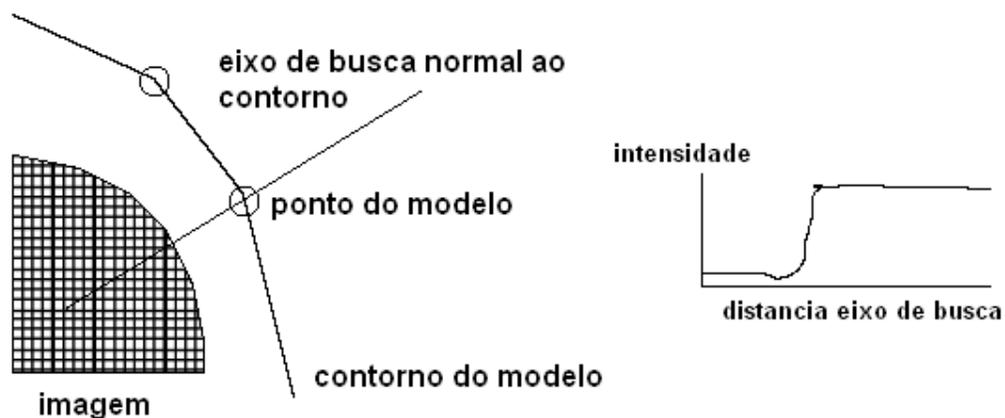


Fig. 17: Busca ao longo do eixo normal ao contorno do modelo.

O sistema mostrado na Figura 17 procura o ponto onde ocorre uma grande mudança na intensidade da imagem. Mas nem sempre os pontos relevantes da imagem estão localizados nos pontos onde ocorrem as maiores mudanças de intensidades. Para ser funcional, o algoritmo deve aprender

como cada ponto se parece, com base em imagens de treinamento. Durante a busca dos pontos na imagem, o modelo é comparado aos pontos do eixo de busca e uma função de custo identifica um novo ponto na imagem, como mostra a Figura 18.

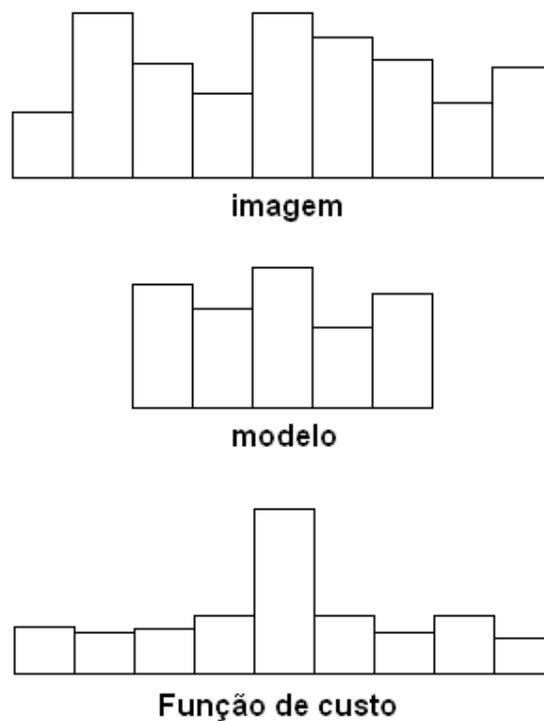


Fig. 18: Função de custo usada para casar o modelo.

Os métodos ASM e AAM são diferentes em alguns aspectos. Enquanto o ASM realiza o casamento dos pontos baseado em um padrão de valores de tons de cinza, o AAM trabalha com um modelo treinado para representar a forma e a textura do objeto a ser procurado. Este modelo pode representar completamente a região circundada pelos pontos. Desta forma a busca pelo objeto no método AAM é feita pela

otimização da diferença entre a imagem representada pelo modelo e o objeto na imagem real.

Um exemplo do uso de modelos flexíveis para detecção de faces é apresentado por Cootes et al. [11]. Neste trabalho a face humana é modelada segundo as informações de bordas e intensidade dos pixels em níveis de cinza.

Com o uso de um banco de dados de imagens de treinamento composto por 40 imagens de faces, o algoritmo construiu um modelo de distribuição de pontos (PDM [10]) que caracteriza uma face como mostrado na Figura 19. Na criação deste modelo são definidos alguns pontos importantes da face, e estes pontos são marcados em um grupo de imagens de treinamento. A posição média e os valores em tons de cinza em volta de cada ponto são usados para formar o PDM. Para realizar a detecção é usado o ASM que tenta ajustar iterativamente a nova face ao modelo PDM.

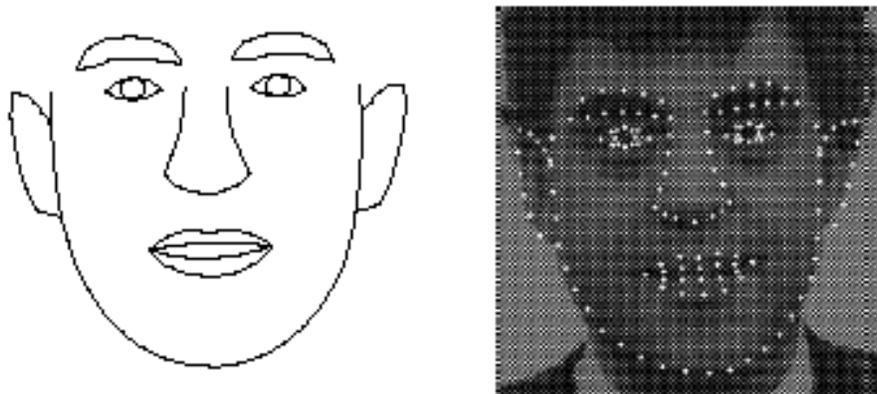


Fig. 19: Modelo PDM de face utilizado por Cootes et al. [11].

2.2. Rastreo de faces

O rastreo de objetos é o nome dado à tarefa de manter atualizada a informação sobre a localização do objeto em uma seqüência de quadros de imagens. Algumas das principais aplicações do rastreo de objetos são análise de imagens médicas, monitoramento de objetos em uma seqüência de imagens e vigilância eletrônica. Para o caso específico de rastreo de faces, uma aplicação possível é nos sistemas de interação entre homem e máquina.

O algoritmo de rastreo de objetos em imagens tem que lidar basicamente com dois problemas: a modelagem do objeto ou ponto de interesse e a previsão de onde o objeto pode estar no próximo quadro. O custo computacional do algoritmo depende fortemente destes dois fatores segundo Bradski [2]. Os métodos de previsão para localização do objeto na próxima imagem podem ser um simples modelo físico de movimento ou um filtro Kalman como o trabalho de Cootes et al. [11].

2.2.1. Métodos Usando Pontos de Interesse.

Uma parte dos métodos de rastreo existentes baseia-se no conceito de Pontos de Interesse, por exemplo [31].

Estes métodos buscam localizar alguns pontos marcantes no objeto e fazem o rastreo através da procura destes pontos nas imagens em seqüência. Em geral, baseiam-se em alguns detectores de características como bordas, cor ou textura, como por exemplo, no trabalho de Harris e Stephens [29].

Definidos os pontos de interesse, o algoritmo tenta encontrar um modelo de trajetória para o ponto fazendo uma análise em alguns poucos quadros. Este modelo de trajetória é aplicado aos quadros subseqüentes para a determinação da nova posição do ponto.

2.2.2. Contornos Elásticos- “Snakes”.

“Snakes” é o nome dado a um método de representar uma curva, que foi introduzido por Kass et al.[21]. Basicamente o “Snakes” funciona como uma seqüência de pontos interligados na imagem, com capacidade de deformar-se para tentar fazer coincidência com alguma borda na imagem.

Para mover os pontos é usada uma função de energia que representa uma força que está sendo aplicado ao ponto. A função de energia depende da posição inicial do contorno e de alguns atributos da imagem (principalmente as bordas que a imagem possui).

Dado uma seqüência de pontos (“Snakes”), o algoritmo tenta iterativamente coincidir os pontos do contorno a alguma borda da imagem, como mostrado na Figura 20.

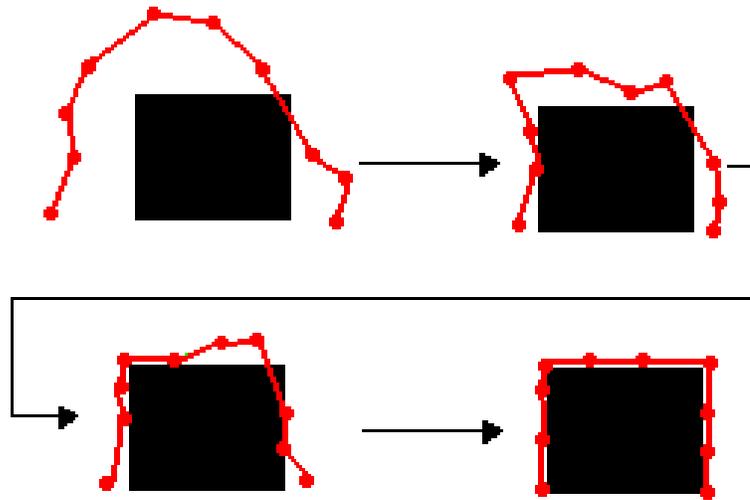


Fig. 20: "Snakes" funcionando.

A tarefa de rastreamento é feita aplicando-se o contorno elástico a cada quadro na seqüência de imagens fazendo a procura da próxima borda, a partir da última posição dos pontos do contorno.

Em uma situação em que o objeto está parcialmente escondido atrás de outro objeto, o algoritmo não consegue encontrar todos os pontos do contorno. Neste caso o histórico das últimas posições dos pontos nos quadros anteriores é usado para formar um modelo que acompanha a parte visível do objeto, até que o mesmo torne-se completamente visível novamente.

2.2.3. Algoritmo CAMSHIFT [2].

O algoritmo chamado CAMSHIFT é usado para rastreamento de objetos em imagens coloridas. Para isto a informação de cor da imagem tem que ser representada como uma distribuição de probabilidades [49], usando histogramas de cor.

No início do processo o algoritmo recebe a informação de localização e o modelo de histograma de cor que representa a cor do objeto. A partir deste ponto o próprio algoritmo faz o cálculo da posição e área do objeto que cada novo quadro, baseando-se na distribuição de probabilidade de cor dentro de uma região de busca. Uma vez calculados localização e tamanho do objeto em determinado quadro, estes dados são usados para a determinação do tamanho da região de busca para o próximo quadro. Através da área calculada para o objeto, é estimada a distância até do objeto até a câmera (Z) e a mudança de rotação do objeto (R).

O modelo de histograma de cor permite a criação da distribuição de probabilidade de cor para a área de busca. O modelo é criado especialmente para a cor do objeto que se deseja rastrear, por exemplo, cor de pele para rastreamento de face. O sistema de cores usado neste trabalho é o HSV (Hue Saturation Value [50]). O modelo de histograma usa o canal H (Hue) do sistema HSV.

O fluxograma da Figura 21 mostra em detalhes as etapas do algoritmo CAMSHIFT.

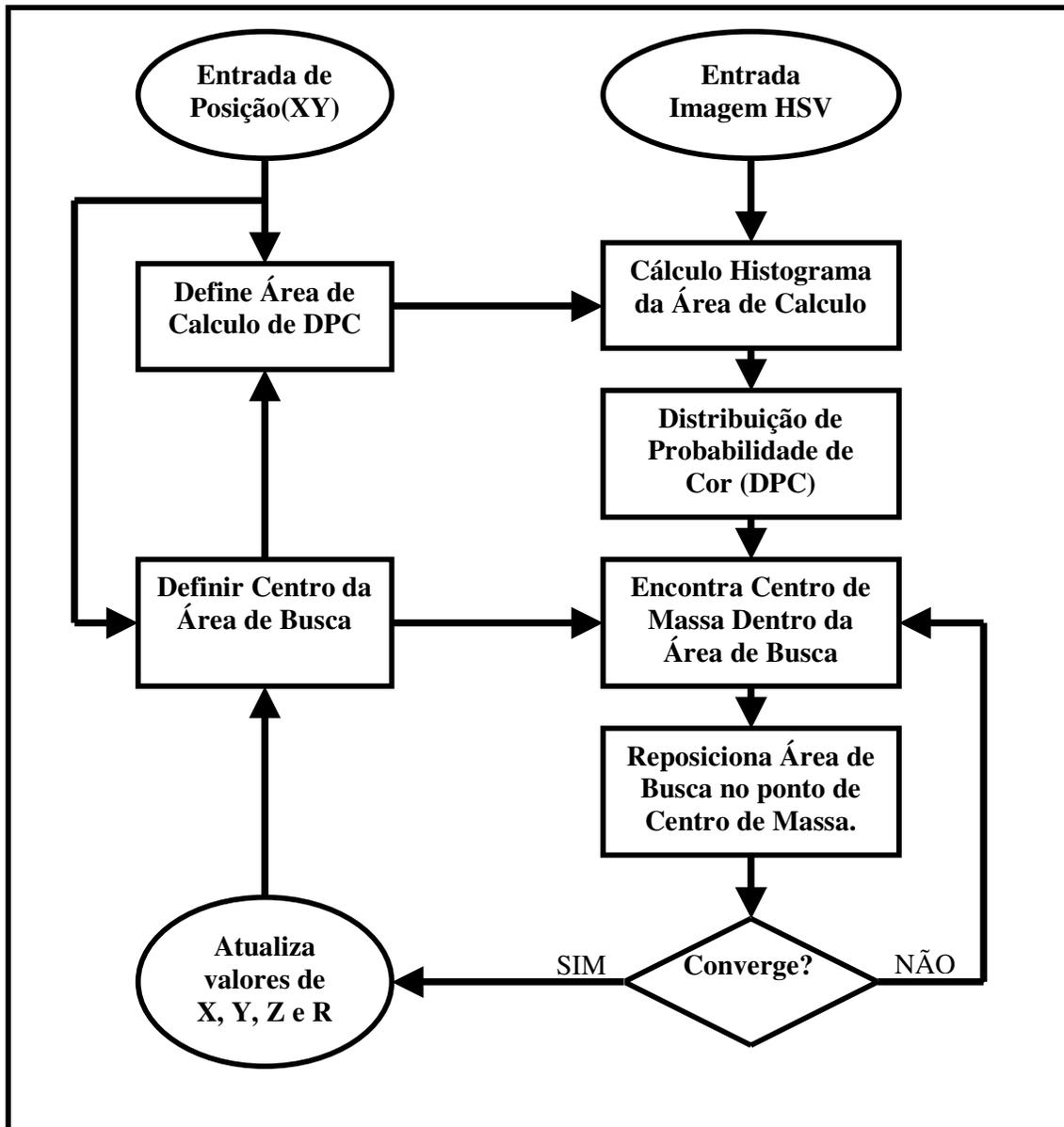


Fig. 21: Fluxograma do algoritmo CAMSHIFT.

A convergência do algoritmo ocorre quando o centro de massa da distribuição de probabilidade de cor dentro da área coincide com o ponto central da área. A Figura 22 mostra dois passos do processo. No passo um a janela da área de busca está deslocada em relação à distribuição. Já no passo dois ocorre a convergência.

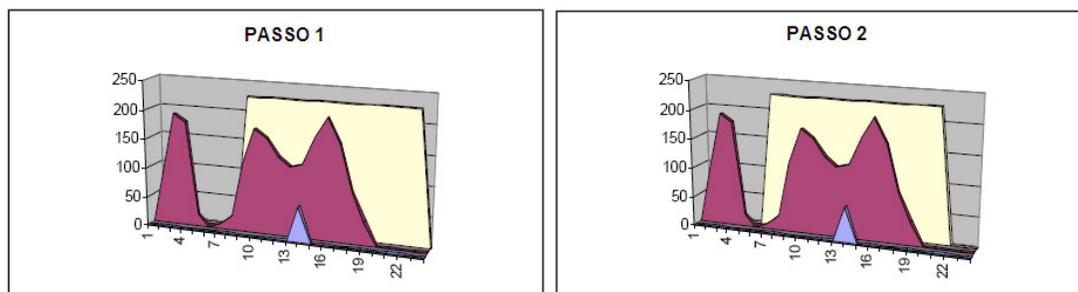


Fig. 22: Método de Convergência do CAMSHIFT.

2.3. Fusão de Informação

Um sistema de fusão de informações é qualquer sistema que recebe dados de entrada externos (sensores) que, baseado em algum tipo de mecanismo de raciocínio, fornece solução para algum tipo de problema ou questão, conforme a definição de Kokar et al. [23].

Segundo Pavlin et al. [29] as Redes Bayesianas possuem características importantes para a criação de um sistema de fusão de informações robusto. Uma destas características é que o processo de inferência na Rede Bayesiana pode ser bastante insensível aos valores dos parâmetros de modelagem usados e pode ter propriedades assintóticas em relação à precisão de classificação segundo Druzdzi e Gaag [12].

Rede Bayesiana é um método de raciocínio baseado em probabilidades que nos últimos anos tem recebido bastante atenção da comunidade de tratamento de probabilidades e incertezas em IA, segundo Charniak [4]. A teoria de Redes Bayesianas tem sido aplicada principalmente em problemas

de diagnósticos médicos, mapas de aprendizagem, compreensão de linguagens e visão computacional [4].

Redes Bayesianas são grafos acíclicos direcionados (DAG's – Directed Acyclic Graph), onde os nós representam as variáveis envolvidas no processo e os arcos que interligam os nós, representam as relações de causa e efeito entre as variáveis do processo, segundo Pearl e Russell [16].

Um dos principais conceitos inerentes ao uso de Rede Bayesiana é a definição de "*Independent Assumption*", que permite executar um raciocínio probabilístico sem ter que definir todas as probabilidades condicionais necessárias em um raciocínio probabilístico puro. Em um sistema qualquer composto de "n" variáveis com dois estados, a combinação total de probabilidades ("*Joint Probabilities*") tem o tamanho exato de $(2^n - 1)$. Em uma Rede Bayesiana podemos considerar somente as relações de dependência mais fortes na hora de definir os arcos que interligam os nós, simplificando o processo de raciocínio [4].

O exemplo da Figura 23 foi proposto por Friedman e Goldszmidt em [51] e serve para o esclarecimento do conceito de "*Independent Assumption*". Neste exemplo, um morador de uma região propensa a terremotos instalou um alarme em sua residência. O alarme pode ser acionado tanto por um terremoto real (0.01) como por um ladrão que invade a casa (0.03). No caso do alarme soar, existe certa probabilidade do vizinho tentar avisar o dono da casa pelo telefone (0.70). Nesta rede temos o nó "*TERREMOTO*" ligado ao nó "*ALARME*" que por sua vez está ligado ao nó "*LIGAÇÃO DO VIZINHO*". O fato de ocorrer um terremoto, indiretamente afeta o fato de ocorrer uma ligação do vizinho. Mas, por outro lado, se estiver disponível a evidência que o alarme foi acionado (ou

não), então o estado do nó “TERREMOTO”, acaba sendo irrelevante neste caso. Assim sendo a Rede Bayesiana assume uma independência entre os nós “*TERREMOTO*” e “*LIGAÇÃO DO VIZINHO*” dado que se sabe o estado do nó “*ALARME*” e não é preciso calcular as probabilidades para a ligação entre os nós “*TERREMOTO*” e “*LIGAÇÃO DO VIZINHO*”.

Uma Rede Bayesiana é definida pela sua estrutura (esquema de nós e arcos) e pelas tabelas de probabilidades que definem as relações de causa e efeito entre as variáveis. Para os nós “raiz” (que não sofrem influência direta de outros nós), temos que definir a probabilidade *a priori* dos possíveis estados de sua variável. Já para os nós intermediários da rede, temos que definir as probabilidades condicionais para suas variáveis. Neste caso é construída uma tabela contendo as probabilidades da variável para cada combinação de estados dos nós precedentes. Por exemplo, um nó que possui dois outros nós “pais”, vai ter em sua tabela quatro valores de probabilidades.

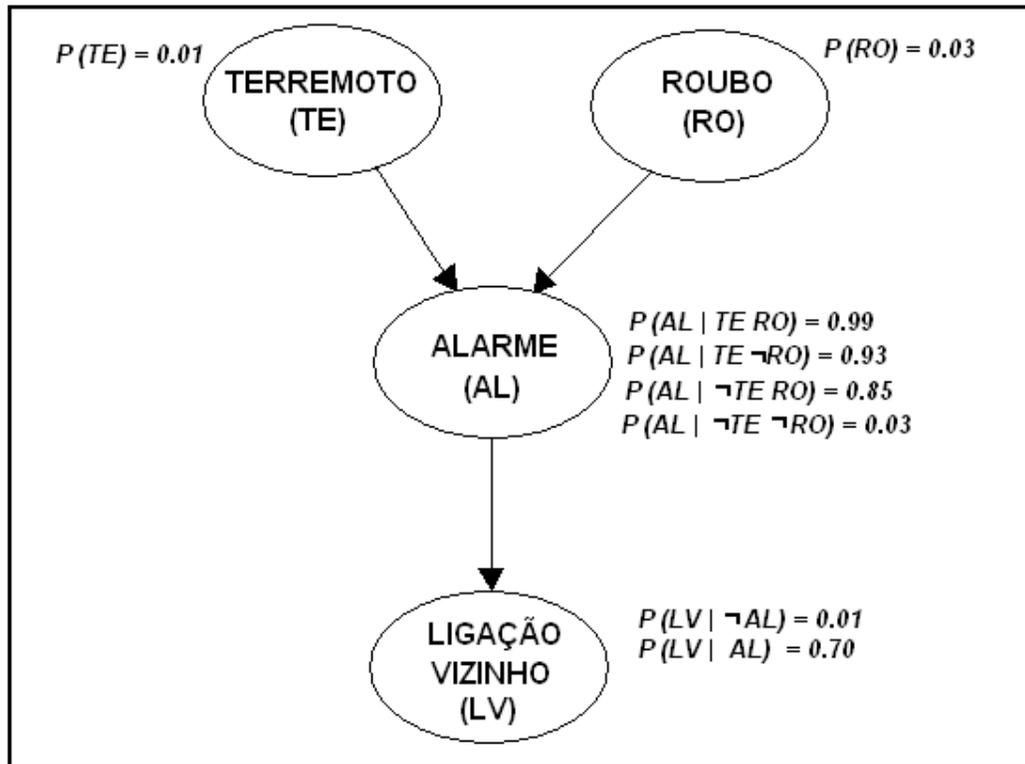


Fig. 23: Exemplo de Rede Bayesiana.

3. MODELAGEM

A modelagem usada neste projeto para o detector de faces baseia-se em quatro elementos. Três destes elementos são os operadores visuais que fazem as buscas por características visuais dentro da imagem. O quarto elemento é a Rede Bayesiana que faz a fusão das informações dos operadores visuais. As respostas dos operadores aplicados às imagens são as entradas (evidências) da Rede Bayesiana que realiza a fusão destas informações fornecendo um valor de porcentagem para o nó "FACE" ser verdadeiro.

A estrutura da Rede Bayesiana que está sendo proposta para este trabalho é mostrada na Figura 24. Na estrutura, a presença ou ausência de face na imagem é representada no nó raiz "FACE". Os nós intermediários "PELE", "BOCA", "OLHO ESQUERDO" e "OLHO DIREITO" representam presença ou ausência destas características faciais dentro das imagens.

Os arcos representam as dependências entre os nós, ou seja, como o estado de um nó interfere na probabilidade de outro. Neste caso, temos o fato de existir uma face na imagem afetando diretamente a probabilidade da presença das características faciais.

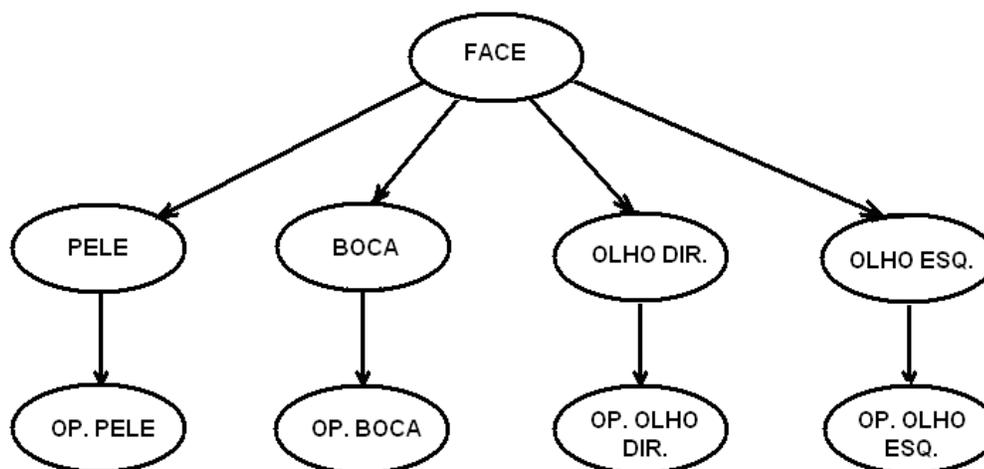


Fig. 24: Estrutura de Rede usada no detector de faces.

No nível mais baixo da rede estão representados os resultados dos operadores de características faciais. Estes nós representam a real capacidade dos operadores para detectar as respectivas características, por exemplo, a probabilidade para o nó “OP.PELE” ser verdadeiro, depende fortemente do estado do nó “PELE”. Este tipo de conexão de rede foi utilizado no trabalho de Marengoni et al. [28].

É importante notar que apenas a dependência entre uma característica e seu correspondente operador está representada na estrutura. Sem dúvida é a dependência mais forte, mas não é a única. Por exemplo, poderíamos representar uma possível dependência entre o nó “OLHO ESQUERDO” e o nó “OLHO DIREITO”. É bem provável que o fato de existir um olho esquerdo na imagem modifica a probabilidade de existir um olho direito.

Por outro lado, como estamos trabalhando com uma rede relativamente pequena, com apenas nove nós, estas conexões secundárias acabam tendo uma influência pequena no resultado final. Por isto estas conexões foram ignoradas neste trabalho.

3.1. Operador Pele

A busca por uma região de cor de pele dentro de uma imagem é um processo simples e computacionalmente barato. Ao mesmo tempo pode fornecer uma forte evidência da presença de uma face na imagem.

O operador pele é baseado em cor. A cor de um pixel pode ser representada em várias formas. Alguns exemplos de sistema de representação de cores são RGB, HSI, CMYK e YCrCb [50]. Para este projeto foi escolhido um modo de caracterizar a cor dos pixels chamado RGB Normalizado. Neste sistema de representação de cores, os valores do pixel são calculados segundo seus valores RGB de acordo com as fórmulas abaixo:

$redN = red / (red + green + blue).$	(2)
$blueN = blue / (red + green + blue).$	(3)

A informação de greenN não precisa ser calculada pois a soma de redN, blueN e greenN é sempre igual a 1.

O sistema de representação de cores RGB Normalizado, tem a característica de eliminar a característica de brilho da cor, preservando a informação de tonalidade.

Durante os experimentos, foi observado que a informação RGB Normalizado dos pixels de cor de pele, quando projetados em um plano, fica concentrada em ao longo da diagonal deste plano. Assim, para fazer a identificação das regiões de pele pela cor, os valores RGB Normalizado são calculados para o pixel da imagem se tornando um ponto no plano RGB Normalizado. Tendo definido o ponto que representa a cor do pixel no plano, então é calculada a distância do ponto para a reta média da faixa de cor de pele. No final a decisão é tomada comparando-se esta distância com o desvio padrão obtido com as imagens de treinamento.

3.2. Operador Olhos

As redes neurais artificiais têm sido usadas para reconhecimento de padrões em imagem digitais com bastante sucesso. Um exemplo de aplicação é o reconhecimento automático de letras e números escritos à mão.

O uso de uma rede neural para a detecção da característica facial “olho” é justificado pela observação que o olho humano apresenta pouca variação em termos de tamanho e forma. No caso específico deste projeto, onde a resolução da imagem e distância entre a pessoa e a câmera, são conhecidos, é perfeitamente possível definir dimensões para uma janela que possa conter as características principais dos olhos para a maioria das pessoas.

Neste projeto, o operador de busca de olhos é uma rede neural artificial de alimentação direta. Esta rede é composta de uma camada de entrada, duas camadas ocultas e uma camada de saída. Os neurônios da rede neural possuem função de ativação do tipo sigmóide, portanto temos em sua saída os valores -1 e $+1$, representando respectivamente os estados “NÃO-FACE” e “FACE”.

O operador funciona do seguinte modo: primeiro uma janela de 15×8 pixels é cortada da imagem e os valores de tom de cinza dos pixels formam um vetor de 120 elementos. Após um processo de normalização que limita os valores deste vetor dentro da faixa entre 1 e -1 , o mesmo é então apresentado à camada de entrada da rede neural.

O desempenho de uma rede neural artificial é fortemente dependente da etapa de treinamento e da estrutura de suas camadas ocultas. Para a escolha da estrutura da rede utilizada foram feitos experimentos com redes de uma e duas camadas ocultas, com cada camada tendo de 25 a 100 neurônios. A estrutura que apresentou melhores resultados foi a que fez uso de duas camadas ocultas, a primeira com 50 neurônios e a segunda com 25 neurônios. A camada de saída possui apenas um neurônio, já que a resposta da rede é apenas “sim” ou “não”.

3.3. Operador Boca

O operador de busca de característica facial boca é um sistema de casamento de modelo. Este sistema foi inspirado na teoria de modelos flexíveis ASM [9], porém para a caracterização dos pontos do modelo, é utilizada a

informação de cor da imagem ao invés da informação em tons de cinza.

Em nosso caso, foi construído um modelo de boca composto de cinco pontos que são usados para realizar a procura pelo contorno da boca na imagem. Ao iniciar uma busca em um determinado ponto da imagem, o modelo se posiciona no formato inicial e seus pontos vão se movendo para fora buscando os contornos da boca.

Após todos os pontos terem encontrado um ponto correspondente em um contorno da imagem, o algoritmo analisa a deformação final do modelo e decide se tem uma boca presente ou não. Na Figura 25(a) pode-se ver a posição inicial e as direções de busca dos pontos que formam o modelo, enquanto que a Figura 25(b) mostra o modelo inicial (em vermelho) posicionado dentro da boca e as posições dos pontos onde o contorno da boca foi encontrado.

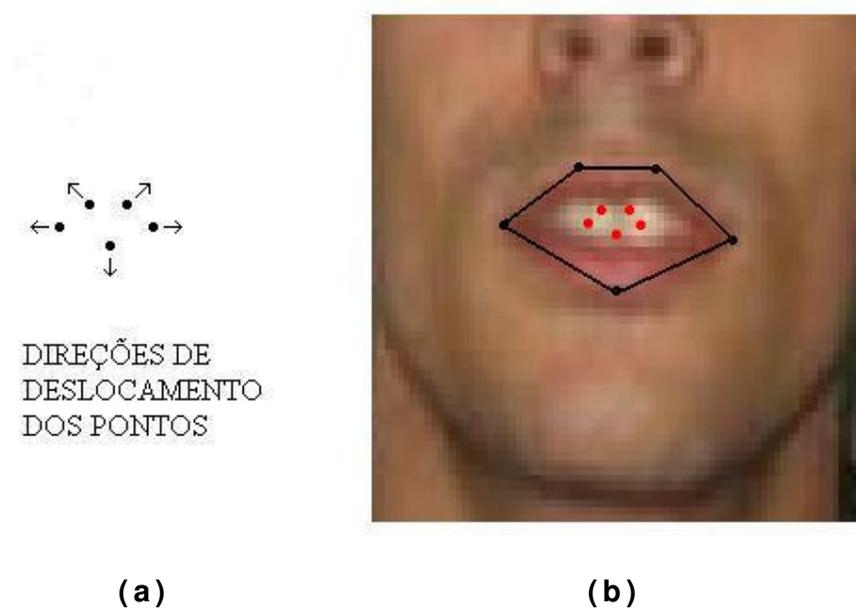


Fig. 25: Modelo auto-ajustável de boca. (a) Modelo inicial. (b) Boca corretamente detectada.

Assim como no operador de busca de pele, a cor do pixel tem uma função importante neste operador. Para determinar se o ponto atingiu um contorno da boca, o algoritmo analisa a cor do pixel e busca uma mudança de cor de lábio para cor de pele. A caracterização da cor é a mesma usada no operador pele, ou seja, RGB Normalizado.

Assim como acontece com a cor de pele, existe uma região no plano RGB Normalizado onde ficam concentrados os pontos captados em regiões com cor de lábio da imagem.

São regiões bem próximas dentro do plano RGB Normalizado, mas conforme os experimentos mostraram, pode-se aproximar estas duas regiões por retas e fazer a classificação de novos pontos pelo cálculo da distância

euclidiana deste ponto para a reta e comparando com um valor de limiar pré-definido.

3.4. Rastreo de Faces

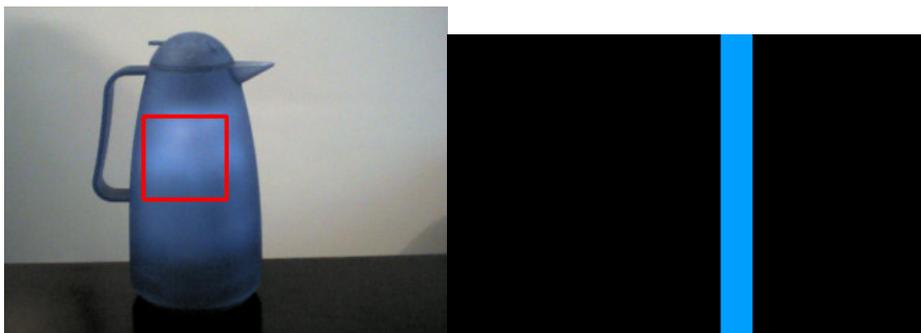
O sistema de rastreo de faces desenvolvido neste trabalho foi inspirado em parte no algoritmo CAMSHIFT visto em detalhes na seção 2.2.3. O modo que este algoritmo realiza o rastreo do objeto permite que ele processe imagens com velocidade de até 30 quadros por segundo, permitindo seu uso em aplicações em tempo real, segundo Bradski [2].

Quando o sistema de rastreo recebe a informação da localização atual da face vinda do detector de faces, é calculado o histograma de distribuição de cor para um quadrado de 60 x 60 pixels com seu centro localizado neste ponto. Este é o padrão de cor que o algoritmo buscará manter durante o rastreo da face. As Figuras 26 mostram alguns exemplos de histogramas criados para quatro objetos diferentes, uma face inclusive. O histograma mostrado à direita de cada imagem foi calculado para a área dentro do quadrado vermelho mostrado em cada imagem.

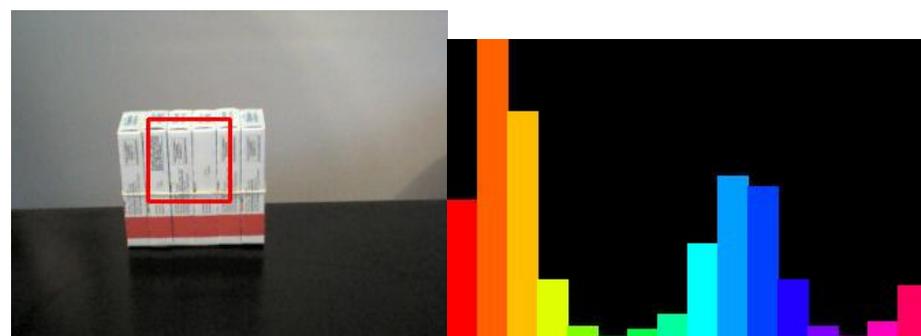
A biblioteca OpenCV possui uma função chamada ***cvCamShift*** que realiza o rastreo pelo algoritmo CAMSHIFT. Esta função recebe a posição do objeto no último quadro e o histograma de distribuição de cor do objeto e retorna a nova posição, área total e orientação do objeto para o quadro atual. O uso desta função proporciona um bom resultado desde que a posição inicial do objeto e o histograma passado estejam corretos. Desta forma, é fundamental que o detector de faces consiga determinar com precisão a posição inicial da face.



(a)



(b)



(c)



(d)

Fig. 26: Imagens com os respectivos histogramas de distribuição de cor.

O modo como o algoritmo faz a atualização da posição da face a cada novo quadro de imagem da câmera está ilustrado na seqüência de imagens da Figura 27.

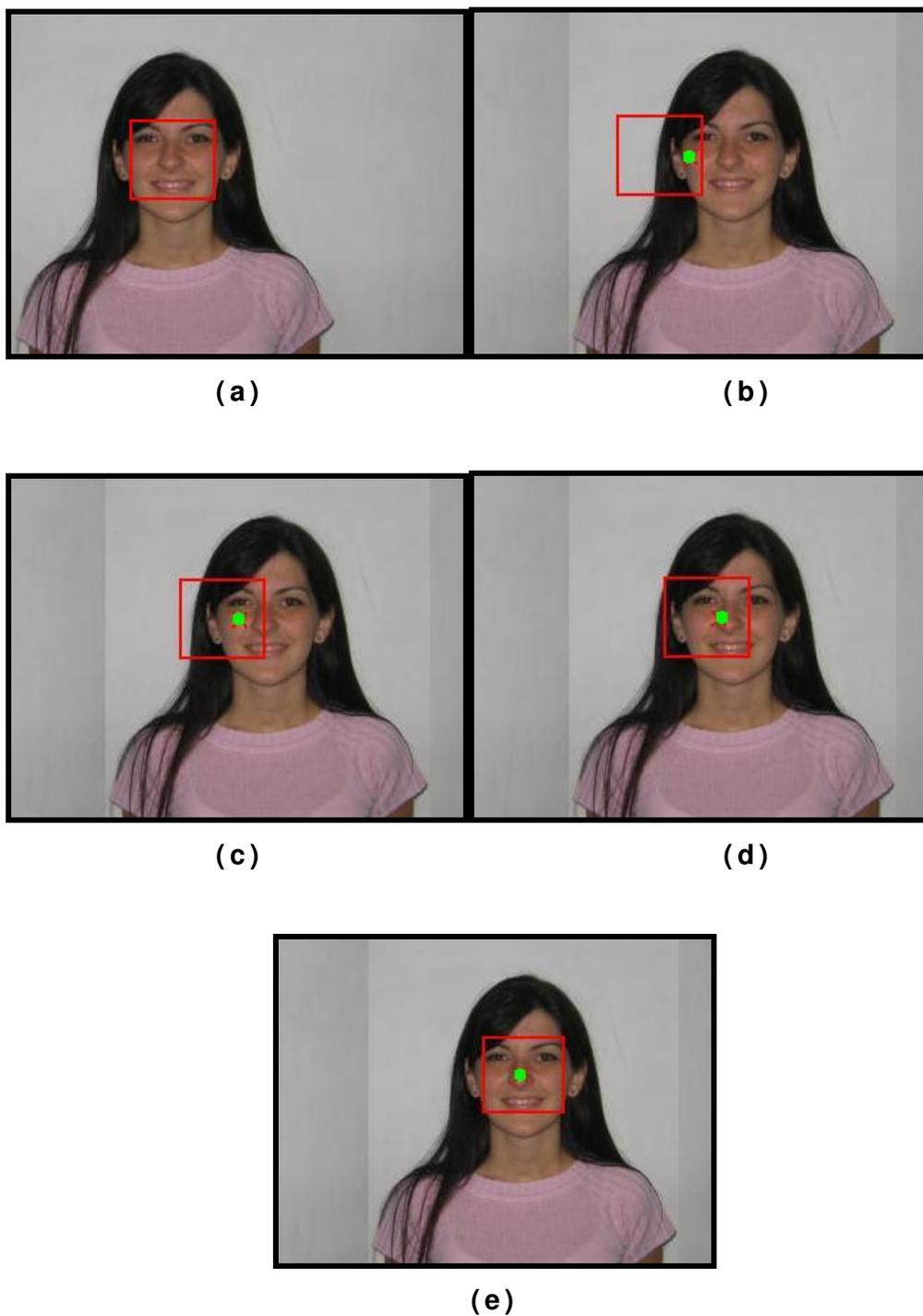


Fig. 27: Seqüência de Imagens para Convergência.

A seqüência inicia na Figura 27(a) em que a posição da face é conhecida. No próximo quadro da seqüência de vídeo, a face moveu-se para a direita da imagem e o algoritmo tem que rastrear a nova posição da face (Figura 27(b)). O quadro vermelho indica a área de busca. Para esta área de busca, o centro de massa da área com cor de pele é indicado no ponto verde. Na seqüência do algoritmo, este ponto é definido como o centro da nova área de busca. Na Figuras 27(c) e 27(d) o processo se repete, até o ponto de convergência do algoritmo mostrado na Figura 27(e). Neste ponto o centro de massa da área com cor de pele coincide com o centro da área de busca.

4. EXPERIMENTOS E RESULTADOS

4.1. Banco de imagens

A parte prática deste projeto iniciou-se com a criação de um banco de imagens para treinamento e teste. Todas as imagens são coloridas e foram tiradas com uma máquina digital CANON modelo A520 com resolução de 640x480 pixels. Em seguida as imagens foram reduzidas pela metade para coincidir com a resolução das imagens geradas por *WEBCAM's* comuns. As imagens do banco de imagens foram divididas em dois grupos, um grupo chamado de "FACE" e outro grupo chamado de "NÃO FACE".

As imagens do grupo chamado "FACE" possuem uma face na posição frontal em cada imagem. As dimensões das faces variam de 73x95 a 94x133 pixels e inclinação máxima de 10 graus em relação ao eixo vertical. Estas características de imagem representam as variações possíveis na aplicação a que se objetiva este detector. Ou seja, estas são as restrições que devem ser impostas às imagens para que o detector funcione com o desempenho esperado.

No grupo "FACE" existe um total de 69 imagens. Estas imagens foram divididas em dois subgrupos. O primeiro chamado de grupo de treinamento, com 43 imagens e o segundo, chamado de grupo de avaliação, composto de 26 imagens. A Figura 28 mostra alguns exemplos de imagens do grupo "FACE".

As imagens do grupo "NÃO FACE" são os exemplos negativos do algoritmo de treinamento e representam o

universo das imagens que não contém nenhuma face. Somam um total de 50 imagens. Estão também divididas em dois subgrupos chamados de grupo de treinamento, com 31 imagens e grupo de avaliação, com 19 imagens.

A necessidade de caracterização do universo de elementos que não pertencem à classe de objetos pertinentes ao detector é um problema importante neste tipo de algoritmo. Neste projeto, o detector de faces é construído para detectar faces em imagens captadas por uma *WEBCAM* dentro de uma sala. Mesmo considerando as restritas possibilidades de aparência que a imagem pode apresentar neste caso, podemos ter como exemplo de “NÃO FACE” os mais variados objetos incluindo mobílias, tecidos e roupas. Representar de forma completa este universo é tarefa bastante difícil. Para minimizar esta dificuldade as imagens “NÃO FACE” do grupo de treinamento apresentam uma grande variação de temas, passando por bonecos inanimados, vasos e perfis de rostos. Alguns exemplos de imagens do grupo “NÃO FACE” são mostrados na Figura 29.



Fig. 28: Exemplos de imagem "FACE".



Fig. 29: Exemplos de imagem "NÃO-FACE".

4.2. Classificação de Cores

O procedimento para a classificação da cor do pixel como pele ou lábio já foi explicado anteriormente (ver seção 3.1).

Para a definição da reta que caracteriza a cor de pele no plano RGB Normalizado, foram utilizadas as imagens do grupo de treinamento de imagens "FACE". Foram calculados os valores RGB Normalizados para 20 pontos de cada imagem. Estes pontos estão indicados na Figura 30, assim como a reta que melhor aproxima estes pontos.

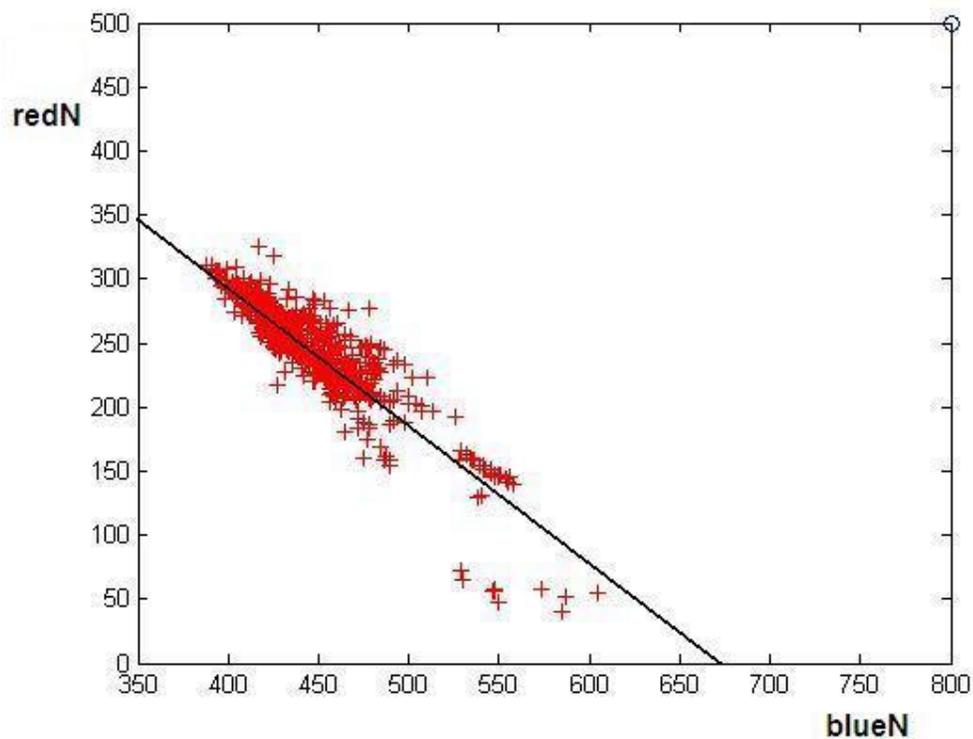


Fig. 30: distribuição dos exemplos de cor de pele no plano RGB Normalizado.

O mesmo procedimento foi utilizado para obtenção das características de cor de lábio dentro do plano RGB Normalizado. Foram calculados os valores RGB Normalizados para 20 pontos localizados em região de lábio de cada imagem do grupo de treinamento de imagens "FACE". Estes pontos estão indicados na Figura 31, assim como a reta que melhor aproxima estes pontos.

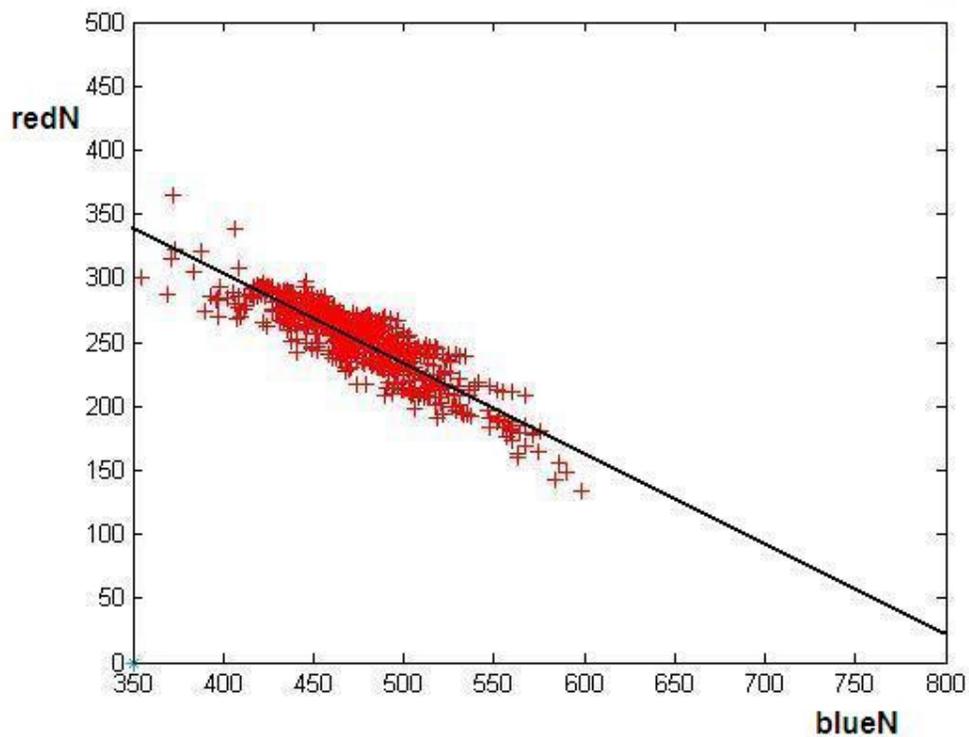


Fig. 31: distribuição dos exemplos de cor de lábio no plano RGB Normalizado.

As equações de reta para cor de pele (4) e cor de lábio (5) foram obtidas utilizando a função *polyfit* do MatLab aplicada aos respectivos conjuntos de pontos. O limiar para classificação do ponto como pertencente à classe pesquisada, foi obtido pelo cálculo do desvio padrão da distância de todos os pontos do conjunto para a reta média.

$\text{redN} = 730,4 - 1.088 * \text{blueN}$	(4)
$\text{redN} = 587,4 - 0,7063 * \text{blueN}$	(5)

4.3. Operador Olhos

Os testes e experimentos com redes neurais, incluindo treinamento e inferência, foram feitos com o pacote de ferramentas para redes neurais do MatLab 7.0 (“*Neural Network Tollbox*”). Durante estes experimentos foram testadas várias configurações de redes e métodos de treinamentos. A configuração de rede que apresentou melhores resultados possui duas camadas ocultas, a primeira com 50 neurônios, e a segunda com 25 neurônios. O MatLab fornece algumas funções para treinamento de redes neurais em seu pacote de ferramentas. A função de treinamento *trainrp* foi a que apresentou melhor convergência neste trabalho. No ANEXO A são encontrados detalhes de operação de todas as funções de treinamento de redes neurais do MatLab.

O procedimento básico para criação da rede neural e seu treinamento começam com um conjunto de imagens contendo 40 imagens “OLHOS”, retiradas de 20 imagens “FACE”, e 40

imagens “NÃO-OLHOS” em tons de cinza. Durante a etapa de treinamento da rede neural é importante que os valores fornecidos estejam dentro de um limite pré-estabelecido. O MatLab fornece algumas funções que transformam linearmente os dados para esta finalidade. Neste trabalho a função usada foi a ***premnmx*** que transforma os dados de forma que fiquem limitados entre os valores -1 e 1 em uma etapa de pré-processamento. No ANEXO A são encontrados detalhes de operação de todas as funções de pré-processamento para redes neurais do MatLab.

A rede neural criada no MatLab, possui alguns parâmetros de configuração, dos quais é importante citar o ***trainParam.goal*** e ***trainParam.lr***. O ***trainParam.goal*** define o valor de erro objetivo para o encerramento de uma sessão de treinamento, enquanto que o parâmetro ***trainParam.lr*** define a taxa de aprendizagem da rede durante o treinamento. Durante os experimentos o parâmetro ***trainParam.goal*** foi definido de forma empírica com o valor 0.1, enquanto que o parâmetro ***trainParam.lr*** foi testado com valores entre 0.05 e 0.2. Nas tentativas de treinamento com o parâmetro ***trainParam.lr*** com valores menores que 0.05 a rede neural não conseguiu convergir. O limite superior de 0.2 foi escolhido para limitar as possibilidades durante os experimentos, uma vez que as redes treinadas com valores maiores não apresentaram resultados melhores.

Após a rede ser criada, foi executada uma primeira etapa de treinamento, tendo como exemplos o conjunto de 40 imagens “OLHOS” e 40 imagens “NÃO-OLHOS”. Após esta etapa, a rede já apresentou um bom desempenho na tarefa de detectar olhos. Por outro lado, esta rede apresenta um alto número de falsas detecções. Isto acontece porque o número

de exemplos negativos foi insuficiente para caracterizar o universo de imagens “NÃO-OLHOS”.

Para solucionar esta deficiência no treinamento, usa-se um método iterativo que busca exemplos de falsas detecções em imagens “NÃO-OLHOS”. Estes exemplos são então acrescentados ao conjunto de treinamento original e uma nova etapa de treinamento é executada. Esta busca é executada por um algoritmo que recebe uma região de imagem que não contem exemplo algum de olho e aplica a rede em todos os pontos desta região. Quando ocorre uma falsa detecção, aquele quadro é escolhido para a etapa de treinamento. Este algoritmo foi inspirado no trabalho feito por Rowley, Bajula e Kanade [32] que já foi explicado anteriormente (ver seção 2.1.1).

Cerca de 20 redes foram criadas e treinadas usando o método descrito. Cada nova rede apresenta um desempenho diferente em função da escolha inicial aleatória dos pesos das ligações entre os neurônios da rede. A maioria delas deixou de convergir após três a seis iterações do processo descrito. Mas outras foram melhorando o desempenho como mostram os gráficos abaixo da Figura 32.

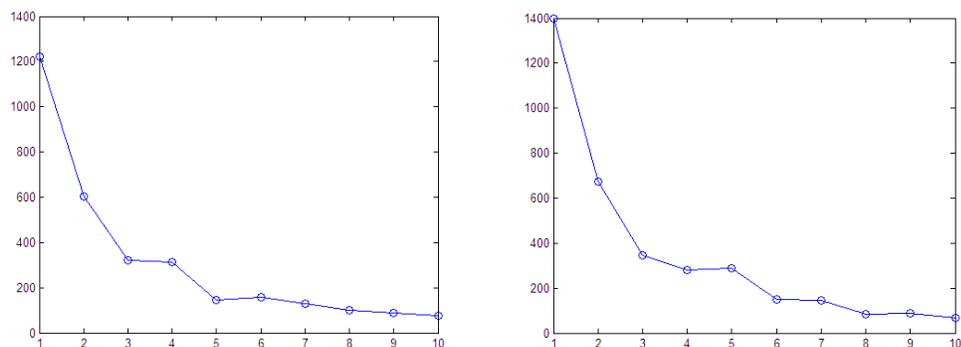


Fig. 32: Exemplos de diminuição no número de falso-positivos durante treinamento de rede neural.

A Figura 33 mostra os resultados do operador de busca de olhos aplicados em duas imagens de teste. Na imagem da esquerda (Figura 33a) o resultado foi correto para ambos os olhos. Na imagem da direita (Figura 33b) o resultado positivo para o olho direito está correto, porém, um falso positivo foi detectado acima do olho esquerdo da pessoa.

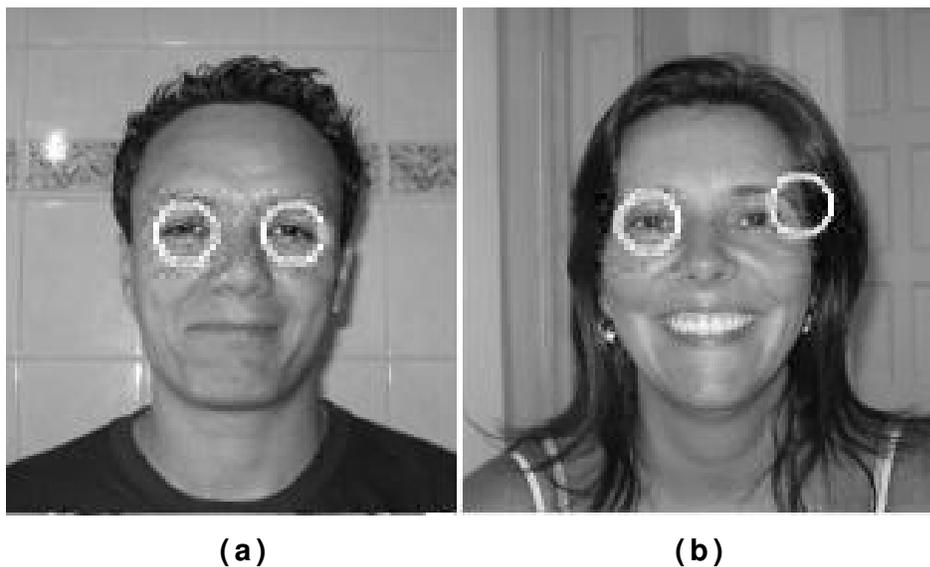


Fig. 33: Resultados do operador de busca de olhos.

4.4. Rede Bayesiana

A parte final do detector de faces é formada por uma Rede Bayesiana que combina os resultados dos operadores de características visuais para fornecer uma resposta confiável, uma vez que os operadores separadamente não fornecem evidência suficiente para o detector de faces.

Neste capítulo será descrito o processo de criação da Rede Bayesiana. Primeiramente será explicado o processo de criação e cálculo das tabelas de probabilidade para os nós da rede. E por último será mostrado como foi feita a etapa de inferência da rede.

4.4.1. Tabelas de Probabilidades

Uma Rede Bayesiana é definida por sua estrutura e pelas tabelas de probabilidades pertinentes a cada nó. Para a criação das tabelas de probabilidades da Rede Bayesiana usada no detector de faces, foram usadas as imagens do conjunto de treinamento que contém 43 imagens “FACE” e 31 imagens “NÃO FACE”.

Nosso detector de faces é construído para detectar faces em imagens captadas por uma *WEBCAM* dentro de uma sala. A probabilidade *a priori* para o nó “FACE” deveria a princípio representar a real ocorrência de eventos “FACE” e “NÃO FACE”, mas não possuímos este dado, em vista da óbvia dificuldade de levantamento.

Neste caso a melhor escolha é usar uma distribuição de probabilidades igualmente dividida:

$P(\text{FACE}) = 50\%$	(5)
$P(\neg\text{FACE}) = 50\%$	(6)

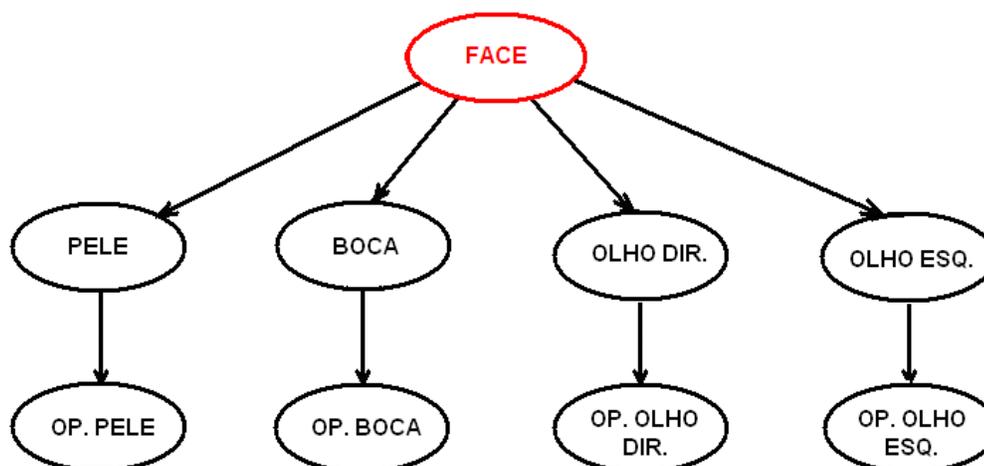
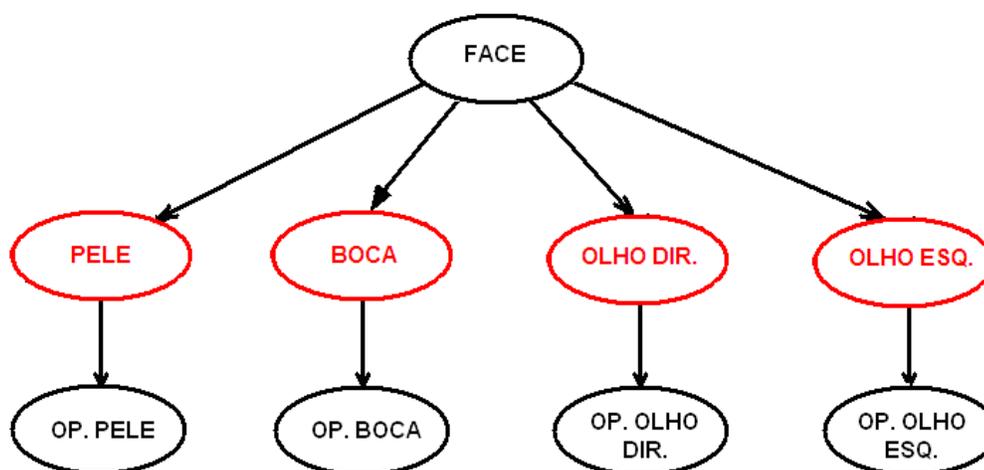


Fig. 34: Detalhe do nó “FACE”.

Para os nós intermediários da rede, o primeiro passo é especificar a quantidade de cada característica facial presente nas imagens do grupo de treinamento. Em certos casos isso pode ser bastante subjetivo, porque em algumas imagens estas características não estão bem definidas. Para contornar isto foi pedido a dez pessoas que analisassem as imagens e dessem suas opiniões a respeito da presença das características dentro das imagens. A Tabela 1 mostra os resultados obtidos.

Tabela 1: Probabilidades para os nós intermediários.

Imagens	Pele	Prob.	OlhoDir	Prob.	OlhoEsq	Prob.	Boca	Prob.
Face(43)	35	35/43	38	38/43	37	37/43	32	32/43
NãoFace(31)	4	4/31	3	3/31	3	3/31	1	1/31

**Fig. 35: Detalhe dos nós referentes à tabela acima.**

O cálculo das probabilidades dos nós “folha” da rede foi feito baseando-se no resultado dos operadores aplicados às imagens de treinamento. Os nós “folha” da rede, que representam os operadores visuais, estão indicados em destaque na Figura 36. A taxa de acerto dos operadores determinou a probabilidade condicional para estes nós. A Tabela 2 mostra os resultados para as 74 imagens do grupo de treinamento. As Tabelas 3, 4, 5 e 6 mostram os resultados separados para cada operador.

Tabela 2: Resultados dos operadores aplicados às imagens do banco de treinamento.

Característica	Total	Resultado Operador	
		Verdadeiro	Falso
Pele	39	33	6
¬Pele	35	6	29
Olho Direito	41	21	20
¬Olho Direito	33	3	30
Olho Esquerdo	40	21	19
¬OlhoEsquerdo	34	3	31
Boca	33	28	5
¬Boca	41	11	30

Tabela 3: Probabilidades para o nó Operador Pele.

Resultado Operador	Presença Pele	Ausência Pele
Verdadeiro	33/39	6/35
Falso	6/39	29/35

Tabela 4: Probabilidades para o nó Operador Olho Direito.

Resultado Operador	Presença Olho Direito	Ausência Olho Direito
Verdadeiro	21/41	3/33
Falso	20/41	30/33

Tabela 5: Probabilidades para o nó Operador Olho Esquerdo.

Resultado Operador	PresençaOlhoEsquerdo	AusênciaOlhoEsquerdo
Verdadeiro	21/40	3/34
Falso	19/40	31/34

Tabela 6: Probabilidades para o nó Operador Boca.

Resultado Operador	Presença Boca	Ausência Boca
Verdadeiro	28/33	11/41
Falso	5/33	30/41

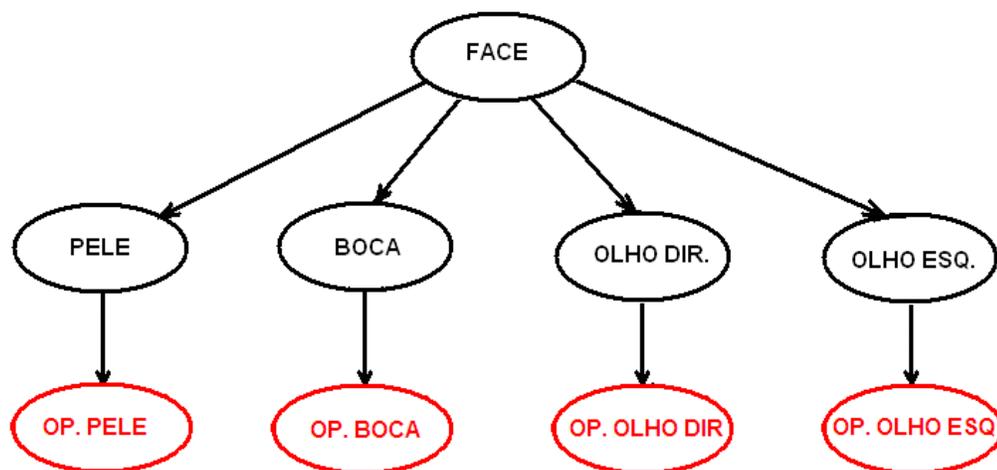


Fig. 36: Detalhe dos nós folha da rede.

4.4.2. Inferência da Rede Bayesiana.

A estrutura de uma Rede Bayesiana, juntamente com as tabelas de probabilidades representa a base de conhecimento previamente introduzida no sistema.

O processo de inferência de uma Rede Bayesiana busca o cálculo da probabilidade de um ou mais nós não observados da rede a partir do conhecimento (evidência) do estado de outros nós.

Os cálculos de inferência da Rede Bayesiana neste projeto, foram feitos com o programa Hugin (*Hugin Expert A/S*) citado em [20].

Para o sistema de detecção de faces deste projeto, o cálculo que se procura fazer é a probabilidade (crença) da presença ou ausência de face na imagem, tendo como entradas as evidências das respostas dos operadores visuais aplicados às imagens. Neste caso são apenas quatro variáveis com dois estados cada, contabilizando dezesseis valores diferentes de probabilidades $P(\text{FACE})$.

Para fazer com que o detector de faces apresente uma resposta mais rápida, o cálculo da probabilidade do nó "FACE" dada uma combinação de respostas dos operadores, é obtida tomando-se o valor correspondente na tabela formada pelos dados obtidos no processo de inferência da rede, executado com o Hugin. A Tabela 7 mostra estes resultados.

Tabela 7: Inferência da Rede para o nó “FACE”.

Op.Pele	Op.OlhoD.	Op.OlhoE.	Op.Boca	P(FACE)	P(¬FACE)
Falso	Falso	Falso	Falso	5,7%	94,3%
Falso	Falso	Falso	Verdadeiro	25,9%	74,1%
Falso	Falso	Verdadeiro	Falso	25,8%	74,2%
Falso	Falso	Verdadeiro	Verdadeiro	66,9%	33,1%
Falso	Verdadeiro	Falso	Falso	25,6%	74,4%
Falso	Verdadeiro	Falso	Verdadeiro	66,6%	33,4%
Falso	Verdadeiro	Verdadeiro	Falso	66,5%	33,5%
Falso	Verdadeiro	Verdadeiro	Verdadeiro	92,0%	8,0%
Verdadeiro	Falso	Falso	Falso	31,0%	69,0%
Verdadeiro	Falso	Falso	Verdadeiro	72,2%	27,8%
Verdadeiro	Falso	Verdadeiro	Falso	72,1%	27,9%
Verdadeiro	Falso	Verdadeiro	Verdadeiro	93,7%	6,3%
Verdadeiro	Verdadeiro	Falso	Falso	71,8%	28,2%
Verdadeiro	Verdadeiro	Falso	Verdadeiro	93,7%	6,3%
Verdadeiro	Verdadeiro	Verdadeiro	Falso	93,6%	6,4%
Verdadeiro	Verdadeiro	Verdadeiro	Verdadeiro	98,8%	1,2%

4.5. *Detector de Faces*

O algoritmo de detecção de face implementado neste projeto está baseado nas bibliotecas de manipulação e tratamento de imagens digitais OpenCV da Intel. A Tabela 8 mostra as etapas do algoritmo criado.

Tabela 8: Etapas do Detector de Face

ITEM	PROCESSO
1	Definição das áreas para cada operador segundo modelo geométrico
2	Aplicação do operador pele.
3	Aplicação do operador boca.
4	Aplicação do operador olho (direito e esquerdo).
5	Busca na tabela de probabilidades segundo resultados dos operadores.
6	Resposta do detector.

A aplicação dos operadores se realiza em todos os pontos dentro da área definida pelo modelo geométrico mostrado na Figura 37. Este modelo geométrico define áreas específicas para aplicação de cada operador (boca, olho direito, olho esquerdo e cor de pele na região da testa).



Fig. 37: Modelo Geométrico da Face.

A resposta final do detector é obtida tomando-se o valor para $P(\text{Face})$ na Tabela 7, de acordo com a combinação de resultados obtidos dos operadores visuais. A resposta é considerada positiva (Face) para valores de $P(\text{Face})$ maiores ou iguais a 66,5%.

A versão final do detector de faces foi testada com as imagens do grupo de avaliação do banco de imagens apresentado no capítulo 5.1. Este grupo de imagens contém 26 imagens de “FACE” e 19 imagens “NÃO-FACE”. Os resultados estão representados na Tabela 9.

Tabela 9: Resultados dos testes do detector de faces com as imagens do grupo de avaliação.

GRUPO	RESULTADO= FACE	RESULTADO= NÃO-FACE	EFICIÊNCIA
26 Imagens Face	23	3	88,4%
19 Imagens Não-Face	0	19	100%

Com a permissão da “*Faculty of Computer and Information Science*” da “*University of Ljubljana, Slovenia*”, foi baixado da internet o banco de imagens “*FRI CVL Face Database*” (<http://www.lrv.fri.uni-lj.si/~peterp/FaceDB.zip>). Este banco de imagens é composto de fotos coloridas tiradas de 114 pessoas, 108 homens e seis mulheres. A Figura 38 mostra alguns exemplos tirados do banco de dados.

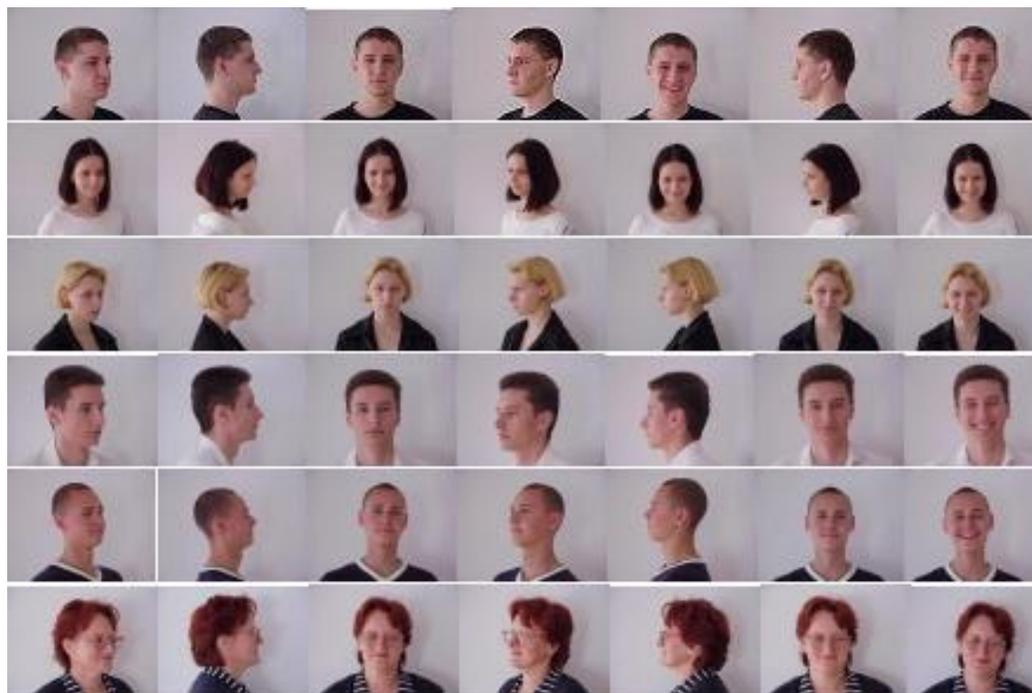


Fig. 38: Exemplos de imagens do “*FRI CVL Face Database*”.

Existem sete fotos de cada pessoa, três frontais completos, dois frontais parciais e dois de perfis. As fotos

foram divididas em três grupos: frontal completo (342 fotos), frontal parcial (228 fotos) e perfil (228 fotos).

O detector de faces foi aplicado nas imagens dos três grupos e o resultado está resumido na Tabela 10.

Tabela 10: Resultados dos testes do detector de faces com as imagens do banco de imagens *FRI CVL Face Database*.

GRUPO	P(FACE)	P(NÃO FACE)
342 FRONTAIS COMPLETOS	299/342= 87,4%	43/342= 12,6%
228 FRONTAIS PARCIAIS	42/228= 18,4%	186/228= 81,6%
228 PERFIS	162/228= 71,0%	66/228= 29,0%

O tempo de execução da versão final do detector de faces é entre 400 e 600 ms. Esta medição foi feita em um computador com processador AMD Athlon (TM)XP2200 1,80GHz. Os operadores de características visuais têm que rodar em uma área de busca definida pelo modelo geométrico, mas quando ocorre uma detecção positiva dentro da área, a busca por aquela característica se encerra e o algoritmo continua rodando. Por causa disto, o tempo de execução total do detector apresenta uma variação de 200 ms.

A Tabela 11 indica os resultados apresentados pelos operadores visuais e pelo detector de faces, aplicados às imagens do grupo de frontais completos.

Tabela 11: Desempenho dos detectores em termos de detecções corretas.

Detector/Operador	Taxa de detecções corretas
Detector de Pele	78,2%
Detector de Boca	75,3%
Detector de Olho	58,9%
Detector de Faces	88,4%

O gráfico da Figura 39 é um histograma do resultado final do teste de avaliação do detector de faces no “**FRI CVL Face Database**”. Ele mostra que 46,8% das imagens alcançaram probabilidade superior a 90% para face presente, o que comprova a qualidade na detecção.

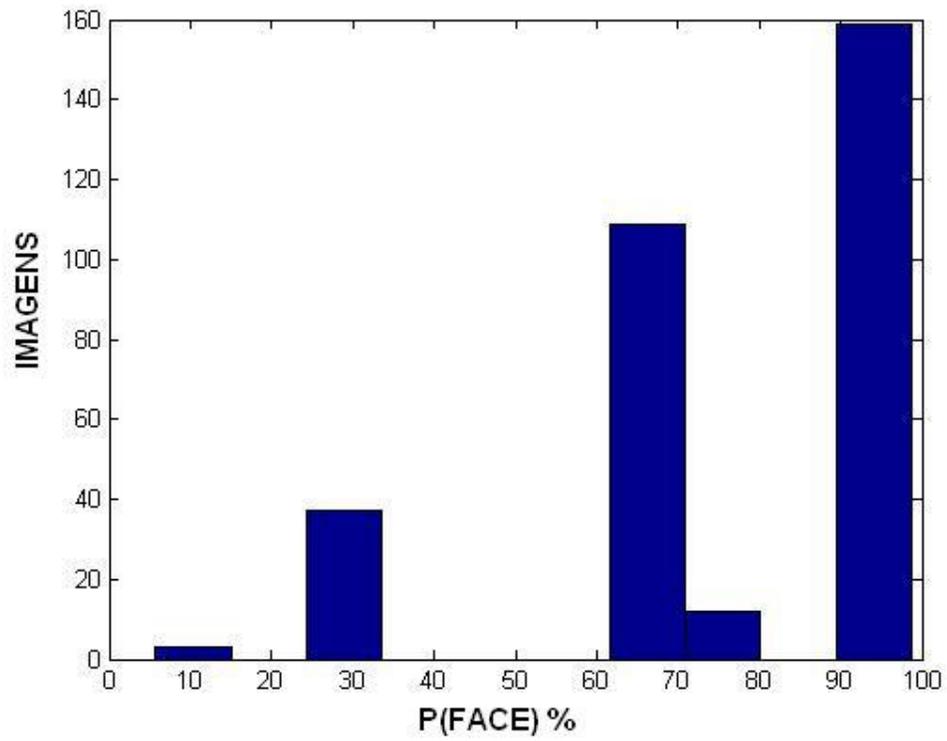


Fig. 39: Resultados do Detector de Faces.

4.6. Rastreo da Face

Uma vez que a face é detectada, entra em operação o algoritmo de rastreo da face. Este algoritmo recebe a posição atual da face na imagem e aplica o algoritmo de rastreo descrito na seção 3.4. Este algoritmo realiza o rastreo da face com base na informação de cor de pele na posição inicial informada pelo detector de face.

O teste prático do rastreador inicia-se com a face da pessoa posicionada no centro da tela do computador e o rastreador conhecendo a posição atual da face. A pessoa começa a mover a cabeça lateralmente em uma velocidade baixa e crescente. Enquanto o rastreador consegue localizar a face dentro da imagem, o algoritmo desenha um círculo vermelho em torno da face. Se dentro de um quadro não for possível para o algoritmo localizar a face, o processo se interrompe e deixa de mostrar o círculo na imagem.

Este experimento foi realizado com taxa de atualização de quadros na câmera de 15, 24 e 30 quadros por segundo. O resultado foi que o rastreador consegue sempre acompanhar os movimentos da face na tela, desde que permaneça visível uma parte da região de pele.

A informação de saída do sistema de rastreo de faces é a posição na tela dos olhos do operador. Esta posição deverá ser passada à próxima camada do controle do mouse ótico, a determinação da visada. O algoritmo de rastreo de face tem a capacidade de fornecer, a cada novo quadro, as dimensões (x/y) da área da face que está sendo rastreada. Com base nestas informações de área da face, um modelo geométrico contendo a posição dos olhos é transformado para coincidir

com as dimensões da face rastreada e a posição dos olhos é determinada. A Figura 40 mostra algumas respostas do sistema de rastreo.



Fig. 40: Resultados do Rastreo de Face.

5. CONCLUSÃO

O trabalho apresentado teve como objetivo principal o desenvolvimento de um sistema de detecção e rastreamento de faces. A aplicação a que se destina este sistema, o mouse ótico, exige que a etapa de detecção apresente uma resposta em tempo real para ser viável.

O tempo de execução medido para o detector de face, em média 500 ms, é suficiente para permitir sua utilização no mouse ótico.

Os experimentos realizados com o detector de faces mostraram que o desempenho do mesmo ficou bem acima da média do desempenho apresentado pelos detectores de características faciais (operadores) individualmente. A Tabela 11 mostra estes resultados.

As pessoas fotografadas no "***FRI CVL Face Database***" possuem pele bem clara, enquanto que o banco de imagens usado para criar o detector apresenta uma variedade maior de tons de pele. Desta forma, a pequena diferença de resultados apresentados, comprova a robustez do detector neste aspecto. O resultado do detector de faces, mostrado na Tabela 11, representa o desempenho no conjunto de imagens criado neste trabalho (26 imagens de faces). O resultado apresentado sobre as imagens do "***FRI CVL Face Database***" foi ligeiramente menor. Do total de 342 imagens frontais, o detector conseguiu identificar corretamente 299 faces, resultando em uma taxa de acertos de 87,5%.

Não estão disponíveis ainda dados sobre outros algoritmos de detecção de faces sendo testado neste banco

de dados, embora ele tenha sido bastante utilizado em testes de algoritmos de reconhecimento de faces.

A Tabela 12, publicada originalmente em [39], mostra o desempenho de alguns algoritmos para detecção de face, em um teste realizado em um conjunto de 26 imagens em tons de cinza contendo 136 faces de tamanhos diversos. O detector de faces desenvolvido neste trabalho não pode ser avaliado com um conjunto de imagens deste tipo, uma vez que a informação de cor é usada no processo. Outro fator que impossibilita a avaliação é o tamanho variável das faces do conjunto de teste. Mas o resultado obtido no conjunto de imagens deste projeto (88,4%) e no “*FRI CVL Face Database*” (87,5%), ficaram bem próximos a media apresentada na tabela 12.

Tabela 12: Desempenho de outros detectores de face aplicados em um conjunto de 23 imagens contendo 136 faces de tamanhos diversos.

MÉTODO	REFERÊNCIA	TAXA DE TECÇÃO	DETECÇÕES FALSAS
Distribution based	[37]	81,9%	13
Neural Network	[32]	90,3%	42
Naive Bayes Classifier	[36]	91,2%	12
Support Vector Machine	[46]	74,2%	20
Mixture of Factor Analyzer	[47]	89,4%	3
SNoW Primitive Features	[48]	93,6%	3

6. REFERÊNCIAS BIBLIOGRÁFICAS

- [1] CANDIDO, Jorge; MARENGONI, Mauricio. Enhancing Face Detection Using Bayesian Networks, *IASTED International Conference on Signal and Image Processing – SIP2006*, v. 1, n. 1, 2006.
- [2] Bradski, G.R. Computer Vision Face Tracking as a Component of a Perceptual User Interface. In Workshop on Applications of Computer Vision, pp. 214-219, 1998.
- [3] CHAIN, Douglas; NGAN, King N. Locating facial region of a head-and-shoulders color image. *Third IEEE International Conference on Automatic Face and Gesture Recognition*, Nara, Japan, p. 124-129, 1998.
- [4] CHARNIAK, Eugene. Bayesian Networks Without Tears. *AI Magazine*, 12(4), p.50-63, 1991.
- [5] CHEN, Qian; WU, Haiyuan; YACHIDA, M. Face Detection by Fuzzy Pattern Matching, *Fifth International Conference on Computer Vision*, p. 591-596, 1995.
- [6] CHENG, Jie; BELL, David; LIU, Weiru. Learning Bayesian Networks From Data: An Efficient Approach Based on Information Theory. *Tech. Rep. – Department of Computer Science – University of Alberta*, 1998.
- [7] CIPOLLA, Roberto et al. Likelihood Models for Template Matching Using the PDF Projection Theorem.
- [8] COOTES, Tim F.; TAYLOR, Chris J.; LANITIS, A. An Automatic face Identification System Using Flexible Appearance Models. *Image and Vision Computing*, v.13, n.5, p.393-402, 1995.
- [9] COOTES, Tim F.; TAYLOR, Chris J.; LANITIS, A. Active Shape Models: Evolution of a Multi-Resolution Method For Improving Image Search. *Proc. Of the 5th British Machine Vision Conference*, v.1, p.327-336, 1994.
- [10] COOTES, Tim F.; TAYLOR, Chris J.; Haslam, J. The Use of Active Shape Models For Locating Structures in Medical Images, *Image Vision and Computing*, v.12, no. 6, p.355-366, 1994.
- [11] COOTES, Tim F.; TAYLOR, Chris J.; EDWARDS, G.J. Active Appearance Models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v.23, p.681-685, 2001.

- [12] COOTES, Tim F.; TAYLOR, Chris J.; EDWARDS, G.J. Learning to Identify and Track Faces in Images Sequences. *IEEE Int. Conference in computer Vision*, pp.317-322,1998.
- [13] DRUZDZL, M.J.; GAAG, L.C. Building Probabilistic Networks: Where do the Numbers come from? Guest editors' introduction. *IEEE Transaction on Knowlegde and Data Engoneering*, pp. 481-486, 2000.
- [14] FREUD, Y; SCHAPIRE, R.E. Experiments with a New Boosting Algorithm, in *Proceedings of the Thirteenth International Conference on Machine Learning*, pp. 148-156, 1996.
- [15] FORESEE, F.D.; HAGAN, M.T. Gauss-Newton approximation to Bayesian regularization. *Proceedings of the 1997 International Joint Conference on Neural Networks*, 1997.
- [16] GILL, P.E.; MURRAY, W.; WRIGHT, M.W. *Practical Optimization*. Academic Press, 1981.
- [17] HARRIS, C.;STEPHENS, M. A combined and corner and edge detector. *Proceedings of 4nd Alvey Vision Conference*, pp. 147-151, 1998.
- [18] INTEL CORPORATION. Open Source Computer Vision Library: Reference Manual. Disponível em <http://www.cs.unc.edu/Research/stc/FAQs/OpenCV/OpenCVReferenceManual.pdf> Acesso em: 20 jul. 2006.
- [19] JENSEN, Finn Verner. *Bayesian Graphical Models. Encyclopedia of environmetrics*, Wiley, Sussex, UK, 2000.
- [20] JENSEN, Finn Verner et al. Hugin – A Shell for Building Bayesian Belief Universes for Expert System. *Proc. Of the Eleventh Intern. Joint Conference on Artificial Intelligence*, Michigan, 1989.
- [21] KASS, M.; WITKIN, A.; TERZOPOULOS, D. Snakes: Active Contours Models. *International Journal of Computer Vision*. v. 1, n. 4. pp 321-331, 1987.
- [22] KERMAD, C.; COLLEWET, C. Improving Feature Tracking by Robust Points of Interest Selection. *Proceedings of the Vision Modeling and Visualization Conference*, pp. 415-422, 2001.
- [23] KOKAR, M.M.; TOMASIK, J.A.; WEYMAN, J.. A Formal Approach to Information Fusion. *Proceedings of the Second Conference on Information Fusion*, pp. 133-140, 1999.

- [24] KUCHI, Prem et al. Human Face detection and tracking using Skin Color Modeling and Connected Component Operators. *IETE Journal of Research*, 2002.
- [25] KUKHAREV, Georgi; MASICZ, Pawel; MASICZ, Piotr. A Fast and Accurate Faces Localization using Gradient Method. *12nd Int. Conf. in Central Europe on Computer Graphics, Visualization and Computer Vision*, p. 149-156, 2004.
- [26] LOWE, D.G.; *Percuptual Organization and Visual Recognition*. Kluwer, Boston, 1985.
- [27] MACKAY, D.J.C. Bayesian interpolation. *Neural Computation*. vol. 4, no. 3, pp. 415-447, 1992.
- [28] MARENGONI, Mauricio et al. Decision Making and Uncertainty Management in a 3D Reconstruction System, *IEEE Transaction on Pattern Analysis and Machine Intelligence*, v. 25, n. 7, p. 852-858, 2003.
- [29] PAVLIN, G.; NINNINK, J.; GROEN, F. Robustness and Accuracy of Bayesian Information Fusion Systems. *Proceedings of Form and Content in Sensor Networks Seminar*, 2006.
- [30] PEARL, Judea; RUSSELL, Stuart; *Bayesian Networks, Technical Report, Handbook of Brain theory and Neural Networks*, MIT Press, 2001.
- [31] RIEDMILLER, M.; BRAUN, H. A direct adaptive method for faster backpropagation learning: The RPROP algorithm. *Proceedings of the IEEE International Conference on Neural Networks*, 1993.
- [32] ROWLEY, Henry A.; BAJULA, Shumeet; KANADE, Takeo, Neural Network-Based Face Detection, *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, p. 203-208, 1996.
- [33] ROWLEY, Henry A.; BAJULA, Shumeet; KANADE, Takeo, Rotation Invariant Neural Network-Based Face Detection, *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, p. 38-44, 1998.
- [34] RUSSEL, S.; NORVIG, P. *Artificial Intelligence: a modern approach*. Prentice Hall, 1995.
- [35] SCHNEIDERMAN, Henry; KANADE, Takeo. A Statistical Method for 3D Object Detection Applied to Faces and Cars. *International Conference on Computer Vision*, 2000.
- [36] SCHNEIDERMAN, H.; KANADE, T. Probabilistic Modeling of Local Appearance and Spatial Relationship for Object

Recognition. Proceedings IEEE Conf. Computer Vision and Pattern Recognition, pp. 45-51, 1998.

[37] SUNG, K.; POGGIO, T. Example-Based Learning for View-Based Human Face Detection. *IEEE Transaction Pattern Analysis and Machine Intelligence*, vol.20, no. 1, pp. 39-51, 1998.

[38] VIOLA, Paul; JONES, Michael J. Robust Real-Time Face Detection. *International Journal Computer*, p.137-154, 2004.

[39] YANG, Ming-Hsuan; KRIEGMAN, David J.; AHUJA, Narendra. Detecting faces in images: A survey, *IEEE Trans. Pattern Anal. Mach. Intell.*, v. 24, n. 1, p. 34-58, 2002.

[40] YANG, Ming-Hsuan; AHUJA, Narendra. Detecting Human Faces in Color Images.

[41] YOW, Kin Choong; CIPOLLA, Roberto. Feature-Based Human Face Detection, *Image and Vision Computing*, v. 15, n. 9, p. 713-735, 1997.

[42] YOW, Kin Choong; CIPOLLA, Roberto. A Probabilistic Framework for Perceptual Grouping of Features for Human Face Detection, *Proc. Of the 2nd International Conference on Automatic Face and Gesture Recognition*, p. 16-21, 1996.

[43] YOW, Kin Choong; CIPOLLA, Roberto. Detection of Human Faces under Scale, Orientation and Viewpoint Variations. *2nd International Conference on Automatic Face and Gesture Recognition*. p. 295-300, 1996.

[44] WU, Haiyuan; YOKOYAMA, Taro; PRAMADIHANTO, Dadet; YACHIDA, M. Face and Facial Feature Extraction From Color Image. *2nd International Conference on Automatic Face and Gesture Recognition*, Killington, p. 345-350, 1996.

[45] WU, Haiyuan; CHEN, Qian; YACHIDA, M. Face Detection from color Images Using a Fuzzy Pattern Matching Method, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 21, n. 1, p. 53-63, 1994.

[46] OSUNA, E.; FREUD, R.; GIROSI, F. Training Support Vector Machines: An Application to Face Detection. Proceedings IEEE Conf. Computer Vision and Pattern Recognition, pp. 130-136, 1997.

[47] YANG, M.H.; AHUJA, N.; KRIEGMAN, D. Mixtures of Linear Subspaces for Face Detection. Proceedings Fourth International Conference of Automatic Face and Gesture Recognition, pp. 70-76, 2000.

- [48] YANG, M.H.; AHUJA, N.; KRIEGMAN, D. A SNoW-Based Face Detector. Advances in Neural Information Processing Systems 12, S.A.Solla, T.K. Leen and K.R. Muller, eds., pp. 855-861, MIT Press, 2000.
- [49] COMANCIU, D.; MEER, P. Robust Analysis of Feature Spaces: Color Image Segmentation. CVPR'97, pp. 750-755, 1997.
- [50] SMITH, A.R. Color Gamut Transform Pairs. SIGGRAPH'78, pp. 12-19, 1978.
- [51] FRIEDMAN, N.; GOLDSZMIDT, M. Building classifiers using Bayesian Networks. Proceedings of the National Conference on Artificial Intelligence. pp. 1277-1284, 1996.
- [52] McCULLOCH, W.; PITTS, W. A logical Calculus of the Ideas Immanent in Nervous Activity. Bulletin of Mathematical Biophysics, Appud KOVACS , 1996.

ANEXO A

FUNÇÕES DO MATLAB PARA CRIAÇÃO E TREINAMENTO DE REDES NEURAIIS

1 – Funções de treinamento de rede.

Trainb – Batch training

A função de treinamento de rede neural *trainb* atualiza os valores de pesos e bias de uma vez após uma rodada completa em que a rede é aplicada em todos os exemplos de treinamento.

Trainbfg – BFGS quasi-Newton Backpropagation

A função de treinamento de rede neural *trainbfg* atualiza os valores de pesos e bias de acordo com o método BFGS Quasi-Newton [1].

Trainbr – Bayesian Regularization.

A função de treinamento de rede neural *trainbr* atualiza os valores de pesos e bias de acordo com o processo de otimização de Levenberg-Marquardt [2] e [3].

Trainc – Cyclical Order Incremental Update.

A função de treinamento de rede neural *trainc* atualiza os valores de pesos e bias a cada apresentação de uma configuração de entradas. As entradas são apresentadas de forma cíclica.

Traingd – Gradient Descent Backpropagation

A função de treinamento de rede neural atualiza os valores de pesos e bias de acordo com o método de gradiente descendente.

Traingdm - Gradient Descent with Momentum Backpropagation.

A função de treinamento de rede neural *Traingdm* atualiza os valores de pesos e bias de acordo com o gradiente descendente com momentum.

Trainrp - Resilient Backpropagation.

A função de treinamento de rede neural *Trainrp* atualiza os valores de pesos e bias de acordo com o algoritmo Backpropagation Resiliente (RPROP)[4].

2 - Funções de pré-processamento.***Premnmx/ Postmnmx***

A função *premnmx* realiza o pré-processamento dos dados de treinamento (entradas e saídas). A função *postmnmx* realiza o pós-processamento da forma inversa. No pré-processamento é feita uma transformação linear dos valores para a faixa entre -1 e 1.

Prestd/ Poststd

A função *prestd* realiza o pré-processamento dos dados de treinamento (entradas e saídas). A função *poststd* realiza o pós-processamento da forma inversa. No pré-processamento os valores são normalizados de forma que o conjunto final tenha media igual a zero e desvio padrão igual a um.

Livros Grátis

(<http://www.livrosgratis.com.br>)

Milhares de Livros para Download:

[Baixar livros de Administração](#)

[Baixar livros de Agronomia](#)

[Baixar livros de Arquitetura](#)

[Baixar livros de Artes](#)

[Baixar livros de Astronomia](#)

[Baixar livros de Biologia Geral](#)

[Baixar livros de Ciência da Computação](#)

[Baixar livros de Ciência da Informação](#)

[Baixar livros de Ciência Política](#)

[Baixar livros de Ciências da Saúde](#)

[Baixar livros de Comunicação](#)

[Baixar livros do Conselho Nacional de Educação - CNE](#)

[Baixar livros de Defesa civil](#)

[Baixar livros de Direito](#)

[Baixar livros de Direitos humanos](#)

[Baixar livros de Economia](#)

[Baixar livros de Economia Doméstica](#)

[Baixar livros de Educação](#)

[Baixar livros de Educação - Trânsito](#)

[Baixar livros de Educação Física](#)

[Baixar livros de Engenharia Aeroespacial](#)

[Baixar livros de Farmácia](#)

[Baixar livros de Filosofia](#)

[Baixar livros de Física](#)

[Baixar livros de Geociências](#)

[Baixar livros de Geografia](#)

[Baixar livros de História](#)

[Baixar livros de Línguas](#)

[Baixar livros de Literatura](#)
[Baixar livros de Literatura de Cordel](#)
[Baixar livros de Literatura Infantil](#)
[Baixar livros de Matemática](#)
[Baixar livros de Medicina](#)
[Baixar livros de Medicina Veterinária](#)
[Baixar livros de Meio Ambiente](#)
[Baixar livros de Meteorologia](#)
[Baixar Monografias e TCC](#)
[Baixar livros Multidisciplinar](#)
[Baixar livros de Música](#)
[Baixar livros de Psicologia](#)
[Baixar livros de Química](#)
[Baixar livros de Saúde Coletiva](#)
[Baixar livros de Serviço Social](#)
[Baixar livros de Sociologia](#)
[Baixar livros de Teologia](#)
[Baixar livros de Trabalho](#)
[Baixar livros de Turismo](#)